SF 298 MASTER COPY KEEP THIS COPY FOR REPRODUCTION PURPOSES

REPORT DOCUMENTATION PAGE

Form Approved
OMB NO. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	January 5, 1998	Final Technical Report 8/95-8/97

4. TITLE AND SUBTITLE

Fast Algorithms for Hybrid Control System Design

5. FUNDING NUMBERS

DAAH04-95-1-0600

6. AUTHOR(S)

Michael D. Lemmon
Panos J. Antsaklis

7. PERFORMING ORGANIZATION NAMES(S) AND ADDRESS(ES)

University of Notre Dame
Dept. of Electrical Engineering
Notre Dame, IN 46556

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

U.S. Army Research Office
P.O. Box 12211
Research Triangle Park, NC 27709-2211

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

ARO 34586.20-MA

11. SUPPLEMENTARY NOTES

The views, opinions and/or findings contained in this report are those of the author(s) and should not be construed as an official Department of the Army position, policy or decision, unless so designated by other documentation.

12a. DISTRIBUTION / AVAILABILITY STATEMENT

Approved for public release; distribution unlimited.

12 b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

Hybrid dynamical systems are systems generating a mixture of continuous valued and discrete event signals. Such systems provide a convenient modeling framework for a variety of complex engineering systems; manufacturing systems, power distribution networks, air traffic control networks, and battefield management systems. Such systems arise frequently in the supervision of complex dynamical processes. This project examined three spects of hybrid system analysis and design. These areas are, 1) multiple model identification, 2) superivsor synthesis, and 3) hybrid system interface design. The objective of this project was to produce computationally efficient algorithms for solving these problems. The project was successful in meetings these goals.

14. SUBJECT TERMS

hybrid systems

15. NUMBER IF PAGES

16. PRICE CODE

17. SECURITY CLASSIFICATION OR REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

Fast Algorithms for Hybrid Control System Design

FINAL TECHNICAL REPORT

MICHAEL LEMMON and PANOS ANTSAKLIS

January 5, 1998

U.S. ARMY RESEARCH OFFICE

GRANT NUMBER DAAH04-95-1-0600

UNIVERSITY OF NOTRE DAME
DEPARTMENT OF ELECTRICAL ENGINEERING
NOTRE DAME, IN, 46556

Approved for Public Release
Distribution Unlimited

TABLE OF CONTENTS

Fast Algorithms for Hybrid Control System Design

Final Report for ARO Grant DAAH04-95-1-0600

Technical Report of the ISIS Group
at the University of Notre Dame
ISIS-98-001
January 5 , 1998

M.D. Lemmon and P.J. Antsaklis
Department of Electrical Engineering
University of Notre Dame
Notre Dame, IN 46556

Interdisciplinary Studies of Intelligent Systems

Fast Algorithms for Hybrid Control System Design

Michael D. Lemmon
Dept. of Electrical Eng.
University of Notre Dame
Notre Dame, IN 46556, USA
(219)-631-8309
Fax:(219)-631-4393
lemmon@maddog.ee.nd.edu

Panos J. Antsaklis
Dept. of Electrical Eng.
University of Notre Dame
Notre Dame, IN 46556, USA
(219)-631-5792
Fax:(219)-631-4393
antsakli@saturn.ee.nd.edu

ARO GRANT: DAAH04-95-1-0600
Final Technical Report: January 5, 1998

Summary

Hybrid dynamical systems (HDS) are systems generating a mixture of discrete-event and continuous-valued signals. Common examples of such systems arise when various continuous systems are interconnected over a computer network. HDS provide a convenient modeling framework for a variety of complex engineering systems; communication networks, manufacturing systems, and distributed battlefield management systems. A pressing need exists for systematic and computationally efficient procedures for the analysis and synthesis of such systems. Between August 1995 and August 1997, the Department of Electrical Engineering at the University of Notre Dame developed computationally efficient methods for the modeling, analysis, and synthesis of hybrid systems under the sponsorship of Army Research Office grant DAAH04-95-1-0600. This project had three significant accomplishments. The first accomplishment developed an interior point method for the identification of multiple models. The second accomplishment developed computationally efficient methods for the synthesis of monitor-type Petri net supervisors. The final accomplishment of this project was the successful application of robust control methods to the synthesis and analysis of switched hybrid systems satisfying uniform ultimate performance bounds.

Research Objectives and Motivation

Hybrid dynamical systems (HDS)[27] are systems generating a mixture of continuous-valued and discrete-event signals. Such systems provide a convenient modeling framework for a variety of complex engineering systems; communication networks, manufacturing systems, power distribution systems, air traffic control networks, and battlefield management networks. While formal modeling frameworks for discrete event or continuous-valued systems have long existed, a combined framework which encompasses both discrete and continuous aspects of such systems has only recently begun to be seriously studied.

Hybrid systems arise frequently in the supervision of complex dynamical processes. In this case, process supervision involves switching the system's structure between various operational modes. When several subsystems can be operated concurrently, then the number of various operational modes grows in an exponential manner with the number of concurrent processes. The resulting system can therefore exhibit a highly complex behaviour which cannot be effectively analyzed using conventional methods. For this reason, computationally effective methods for modeling, analysis,

and synthesis of hybrid systems must be developed. The objective of this project was to develop computationally efficient methods for hybrid system design and analysis.

Technical Approach

The computational algorithms developed in this project work within the context of the Antsaklis-Stiver-Lemmon (ASL) [28] framework for HDS modeling. This framework reconfigures general HDS into a discrete-event and continuous-state system which communicate through an interface. HDS synthesis is then viewed as a two step process. The first step constructs a high-level discrete-event system (DES) supervisor. The second step synthesizes an interface executing the supervisor's commands in a "safe" and "optimal" manner.
The value of this two-step appraoch is that the required synthesis problems can now be formulated as a generalization of well understood analog and DES controller design problems. On a theoretical level, this approach makes it easier to understand the fundamental properties which are characteristic of HDS. On the practical side, this strategy allows us to draw on a variety of mature and stable numerical procedures in solving these problems. In this project, we have focused on the numerical implementation of various HDS design methods. Our work has focused on three specific subproblems encountered in HDS design;

1. multiple-model identification,
2. supervisor synthesis,
3. and the design of safe and optimal HDS interfaces.

The technical approach being pursued in each of these subproblems is briefly summarized below.

Multiple-Model Identification: The objective is to extract a set of linear dynamical models from the observed input/outputs of an unknown system. The identification problem involves determining what the linear dynamical models are and over what range of inputs the model should be used. This identification problem is framed as a constrained optimization problem that is equivalent to coding problems possessing a "fidelity" criterion. Expectation-Maximization (EM) procedures provide one popular method for solving this type of problem. This is the approach followed by some of the neural network groups (Jordan/Jacob) in which an ad hoc class of algorithms known as "alternating minimization" (AM) have been employed. Our work has used interior-point (IP) optimization methods to implement an incremental version of the EM-algorithm. This class of procedures is mathematically well understood and we've shown that EM-procedures based on IP methods can exhibit the same computational efficiency enjoyed by IP methods used for linear programming.

HDS Interface Synthesis: HDS synthesis requires the determination of an interface that executes the supervisor's commands in a "safe" and "optimal" manner. Safety refers to the invariance of discrete event traces to small variations in HDS state. Optimality refers to how well the executed commands are carried out. We've been able to solve this problem for switched systems of linear parameter varying plants using linear robust control methods. Due to the safety constraint, it is important that synthesized controllers satisfy a uniform ultimate bound (bounded amplitude) on their performance variables. We've developed methods based on popular linear matrix inequality techniques to synthesize and analyze switched output feedback control systems satisfying an induced $\mathcal{L}_\infty$ performance criterion.

Petri Net Supervsior Synthesis: Our earlier work synthesized DES supervisors under the assumption that the plant's symbolic behaviour could be represented by a non-deterministic finite state machine (FSM). In many cases, a vector FSM or Petri net (PN) representation of the plant's symbolic behaviour is more appropriate. We have developed a matrix-based approach to the synthesis of Petri-net supervisors similar to controllers used by Giua and Lewis. Our method involves determining additional controller places so the controlled network possesses specified place invariants. The synthesis procedure is computationally efficient and well-suited to on-line controller reconfiguration. We've been able to characterize the class of feasible controls for Petri nets with unobservable transitions and have applied this methods to deadlock avoidance problems.

Significant Accomplishments and Activity

This project was a single year project which originally expired in August of 1996. At that time a no-cost extension of the project was requested to follow up on some of the signficant results which were generated by the grant.
Major accomplishments for this project are;

- the development of the hybrid interior point (HIP) algoirthm for multi-model identification,
- the development of robust approaches to the design of hybrid control system interfaces,
- and extending our Petri net synthesis work to plants with uncontrollable and unobservable transitions.

The significant accomplishments in each of these areas is summarized in the following subsections.

HIP Algorithm: Our work [1] [2] with the HIP algorithm has yielded rigorous proofs characterizing the algorithm's asymptotic behaviour and complexity properties have been completed. The use of this algorithm in the efficient training of radial basis function (RBF) neural networks was demonstrated. Under this grant a "large-step" version of the algorithm was developed. This algorithm exhibited a computational cost, scaling, and accuracy which significantly outperformed existing implementations of EM-algorithms. The technical report on this work will be found in the attached preprint of a journal article that was accepted by the journal "Neural Networks" for publication. Under this part of the project a graduate student (Dr. P. Szymanski) completed his doctoral dissertation.

HDS Interface Design: Important progress was made in the design of safe interfaces for hybrid dynamical systems. The formulation of this problem has been developed over a series of papers [3]-[13]. The prinicpal results of these studies will be found in [6], [7], [8], and [10]. Reference [6] characterizes linear parameter varying state feedback control systems which satisfy a uniform ultimate boundedness specification. Reference [7] shows how the results in [6] can be used to design hybrid system interfaces that are "safe" in the sense that they do not violate guard conditions. Reference [8] extends the results of [6] and [7] to show how these methods can be used to extract timed automata models for switched linear systems. Reference [10] generalizes the results in reference [6] to output feedback and self-scheduled bounded amplitude control. These results, taken as a whole provided a systematic and computationally efficient basis upon which to design families of controllers for switched dynamical systems. Some important extensions to these results have recently been obtained. Motivated by our earlier work in invariant based design of hybrid systems [14] [30], we examined hybrid automata and Petri net models of hybrid systems [15]. These studies showed that the methods in [7] and [8] could be used to assess the bounded amplitude performance

of hybrid systems represented by hybrid automata [16] or hybrid Petri net [17]. Our future work will continue along this line of inquiry. This component of the project has supported the doctoral work of a graduate student (C.J. Bett) who will be completing his dissertation in Spring 1998. This work also partially supported the work of a Masters level student (K.X. He) who will also be defending his master's thesis in Spring 1998. Copies of references [6],[7], [8], [10], [16], and [17] have been attached in the appendix.

Petri Net Controller Synthesis: Significant progress was made in extending our matrix-based PN-synthesis methods [29] to process nets with uncontrollable and unobservable transitions. These results expand the class of problems which can be addressed to those involving controllable and unobservable transitions. Early work in this area [29] provided a computationally efficient method for designing feedback Petri net controllers. Extensions of this work have been reported in [18]-[26]. A summary of this work will be found in [24] which has been attached in the appendix. Important applications of this method to the problem of deadlock avoidance will be found in [25] which is also attached in the appendix. This part of the project supported the Ph.D. work of one graduate student (J. Moody) whose completed his dissertation in December 1997.

List of Publications and Technical Reports

The following technical reports were generated by this grant.

1. Szymanski, P.T., Lemmon, M.D.,and Bett, C.J. (1996), " Hybrid Interior Point Training of Modular Neural Networks to appear in *Neural Networks.* Also released as Technical Report of the ISIS Group at the University of Notre Dame, Report:ISIS-96-007, December 1996.

2. Szymanski, P.S., Lemmon, M.D., (1996), "Alternating Minimization Training of Radial Basis Function Networks", Proceedings SPIE conf. on applications and science of artificial neural networks, Orlando Florida, April 1996. (invited paper)

3. Lemmon, M.D. and Bett, C.J. (1995), "hybrid control system design using robust linear control agents", IEEE Conference on Decision and Control, December 1995, New Orleans, LA

4. Bett, C.J. and Lemmon, M.D., (1996), "H-infinity Gain Schedule Synthesis of Supervisory Hybrid Control Systems", Hybrid Systems III, Springer Verlag, 1996. Also presented at the DIMACS/SYCON conference on Hybrid systems October 1995.

5. Lemmon, M.D. and Bett, C.J. (1996), "robust hybrid control system design", Proc. of the IFAC 13th World Congress, San Francisco, July 1996.

6. C. Bett and M. Lemmon, "Finite-Horizon Induced L-infinity Performance of Linear Parameter Varying Systems", Technical Report ISIS-96-005 of the ISIS Group at the University of Notre Dame. also to appear in 1997 American Control Conference. Submitted for publication to *Automatica*

7. M. Lemmon and C. Bett "Verification of Safe Inter-Event Behaviour in Supervisory Hybrid Dynamical Systems", Preprint presented to the Hybrid Systems and Autonomous Control Workshop, Cornell University, Oct. 1996. Released as Technical Report, ISIS-96-006, of the ISIS group at the University of Notre Dame. to appear in *International Journal of Control.*

8. C.J. Bett and M.D. Lemmon, Bounded Amplitude Performance of Switched LPV Systems with Applications to Hybrid Systems, Submitted to *Automatica*, September 1997. Expanded version of technical report ISIS-97-004, Dept. of Electrical Eng., University of Notre Dame, March 1997.

9. C.J. Bett and M. D. Lemmon, "On Linear Fractional Representations of Multidimensional Rational Matrix Functions", ISIS Technical Report isis-97-008, June 1997

10. C.J. Bett and M.D. Lemmon, "Sufficient Conditions for Self-Scheduled Bounded Amplitude Control", September 1997. ISIS Technical Report ISIS-97-009, submitted to *Automatica* as brief paper.

11. M.D. Lemmon and C.J. Bett, Safe Implementations of Supervisory Commands, In Hybrid Systems IV, Antsaklis, Kohn, Nerode, and Sastry (editors), Springer Verlag, LNCS 1273, pp. 235-247, 1997

12. M.D. Lemmon and C.J. Bett, Extracting Timed Automatons from Switched Linear Paramter Varying Plants, to appear in proceedings for the fifth international workshop on hybrid systems, Sept 11-13, 1997, University of Notre Dame, Indiana.

13. C.J. Bett and M.D. Lemmon, Finite-horizon induced $\mathcal{L}_\infty$ performance of linear parameter varying system, In *Proceedings of the American Control Conference*, Albuquerque, NM, June 1997.

14. James Stiver, P. Antsaklis, M. Lemmon, "an invariant based approach to the design of hybrid control systems", Proc. of the IFAC 13th World Congress, San Francisco, July 1996.

15. M.D. Lemmon and P.J. Antsaklis, Timed Automata and Robust Control: can we now control complex dynamical systems?, IEEE Conference on Decision and Control, San Diego, California, December 1997.

16. K.X. He and M.D. Lemmon, "Lyapunov Stability of Continuous-valued Systems under the Supervision of Discrete-Event Transition Systems", ISIS Technical Report ISIS-97-010, September 1997, to appear in Hybrid Systems: computation and control, International Workshop, Berkeley California, USA, April 13-15, 1998.

17. M.D. Lemmon, K.X He, and C.J. Bett, Modeling Hybrid Control Systems using Programmable Timed Petri Nets, to appear in L'Automatisation des Processus Mixtes (ADPM'98), Rheims, France, March 19-20, 1998.

18. J.O. Moody, P.J. Antsaklis and M.D. Lemmon, "Feedback Petri Net Control Design in the Presence of Uncontrollable Transitions", Proc of the 34th IEEE Conference on Decision and Control, pp. 905-906, New Orleans, Louisiana, Dec 13-15, 1995.

19. J.O. Moody, P.J. Antsaklis and M.D. Lemmon, "Automated Design of a Petri Net Feedback Controller for a Robotic Assembly Cell", INRIA/IEEE Conference on Emerging Technologies and Factory Automation, Paris, France, October 10-13, 1995.

20. J.O. Moody, P.J. Antsaklis, and M.D. Lemmon, (1996) "Petri Net Feedback Controller Design for a Manufacturing System", Proc of the IFAC 13th World Congress, San Francisco, July 1996.

21. J.O. Moody and P.J. Antsaklis, "Petri Net Supervisors for DES in the Presence of Uncontrollable and Unobservable Transitions", Proceedings of the 33rd Annual Allerton Conference on Communications, Control and Computing, pp. 176-185, Allerton House, Monticello, IL, October 4-6, 1995. Also released as ISIS Technical Report isis-97-012, October 1997.

22. J.O. Moody and P.J. Antsaklis, "Supervisory Control of Petri Nets with Uncontrollable/Unobservable Transitions", Proc of the 35th IEEE Conference on Decision and Control, pp.4433-4438, Kobe, Japan, Dec 11-13, 1996. Also released as ISIS Technical Report isis-97-013, October 1997.

23. J.O. Moody and P.J. Antsaklis, "Characterization of Feasible Controls for Petri Nets with Unobservable Transitions", Proceedings of the 1997 American Control Conference, pp. 2354-2358, Albuquerque, New Mexico, June 4-6, 1997. Also released as ISIS Technical Report isis-97-014, October 1997.

24. J.O. Moody and P.J. Antsaklis, "Supervisory Control Using Computationally Efficient Linear Techniques: A Tutorial Introduction", Proc of 5th IEEE Mediterranean Conference on Control and Systems, Session MP1, Paphos, Cyprus, July 21-23, 1997. Also released as ISIS Technical Report isis-97-015 October 1997.

25. J.O. Moody and P.J. Antsaklis, "Deadlock Avoidance in Petri Nets with Uncontrollable Transitions", ISIS Technical Report isis-97-016, October 1997.

26. X.D. Koutsoukos and P.J. Antsaklis, "An Approach to Hybrid Systems Control Applied to Clocks," Proceedings of Hybrid Systems V, pp. 137-144, University of Notre Dame, Notre Dame, IN, September 11-13, 1997.

27. *Hybrid Systems IV*, P. Antsaklis, W. Kohn, A. Nerode, S. Sastry Eds., 405 pages, Lecture Notes in Computer Science, LNCS 1273, Springer-Verlag, 1997.

28. J.A. Stiver, P.J. Antsaklis and M.D. Lemmon , "A Logical DES Approach to the Design of Hybrid Control Systems", Math. and Computer Modeling, Special Issue on Discrete Event Systems, pp. 55-76, June 1996.

29. K. Yamalidou, J. Moody, M. D. Lemmon and P. J. Antsaklis, "Feedback Control of Petri Nets Based on Place Invariants", Automatica, Vol 32, No 1, pp 15-28, January 1996.

30. J.A. Stiver, P.J. Antsaklis and M.D. Lemmon, "Hybrid Control System Design Based on Natural Invariants", Proc of the 34th IEEE Conference on Decision and Control, pp. 1455-1460, New Orleans, Louisiana, Dec 13-15, 1995.

Awards and Honors

- Dr. Panos Antsaklis,
  - IEEE Distinguished Lecturer, Control Systems Society, 1996-.
  - Honorary Chair of the 4th IEEE Mediterranean Symposium on Control and Automation, Crete, Greece, June 10-14, 1996.
  - President IEEE Control Systems Society (CSS), 1997.
  - Keynote and Plenary Speaker at Conferences

* Annual Rockwell Control Conf. and Power Seminar of Allen-Bradley, Mequon, WI. May 3, 1996.
* Adv. Summer Inst. ASI'96, the Annual Conference of the European Network of Excellence in Intelligent Control and Integrated Manufacturing Systems
* (ICIMS), Toulouse, France, June 2-6, 1996.
* ISIC Plenary Address at the IEEE CCA/ISIC/CACSD conference on Control Applications (CCA), Intelligent Control (ISIC), and Computer Aided Control System Design (CACSD), Dearborn, Michigan, September 17, 1996.
* International Conference on Intelligent Systems: A Semiotic Perspective, National Institute of Standards and Technology (NIST), Gaithersburg, Maryland, October 23, 1996.
* International Symposium on Issues and Challenges of Manufacturing Education for the 21st Century, University of Patras, Greece, June 26, 1997.

– Guest Editor

* Special issue on "Hybrid Systems", IEEE Transactions on Automatic Control, April 1998.
* Special issue on "Hybrid Systems", Journal on Discrete Event Dynamic Systems, 1998.

– Invited Panelist

* Panel Discussion on "Control, Automation: Friends or Foes?" at the 4th IEEE Mediterranean Symposium on Control and Automation, in Crete, Greece, June 10-14, 1996.
* Panel Discussion on "Intelligent Control Systems" at the 13th Triennial World Congress of the International Federation of Automatic Control, San Francisco, CA, July 1-5, 1996.
* Panel Discussion on "Intelligent Systems: Beyond Neural Networks and Fuzzy Control" at the 13th Triennial World Congress of the International Federation of Automatic Control, San Francisco, CA, July 1-5, 1996.

- Dr. Michael Lemmon,
  - Chair of IEEE Working Group on Hybrid Systems
  - Editor for electronic newsletter for the IEEE Working Group on Hybrid Systms
  - Program chair, Fifth International Workshop on Hybrid Systems, Notre Dame, 1997.
  - Program Chair, 1998 International Symposium on Intelligent Control
  - Operating Committee, 1998 IEEE Conference on Decision and Control
  - Member of IEEE Control System Society Conference Editorial Board
  - Program Committee, 1998 American Control Conference
  - Invited Speaker, Workshop on "Discrete Event and Hybrid Systems", IEEE Robotics and Automation Conference, April 20-25, 1997, 1997, Albuquerque New Mexico.
  - Invited Speaker at special session on "Intelligent Control Systems" at the 3rd Joint Conference on Information Sciences, Durham, North Carolina.
  - Guest Editor for special issue on "Neural Networks in Control" for the IEEE Transaction on Neural Networks, est. publication date 1999.

– Guest Edtior for special issue on "Hybrid Systems", Journal on Discrete Event Dynamic Systems, 1998.

Graduate Students Supported

4 graduate students were partially supported with this grant.

Number of MS and Ph.D. Degrees Awarded

2 Ph.D. was awarded to students supported by this grant. 1 additional Ph.D. will be awarded in Spring 1998 to a student supported by this grant.

List of Scientific Personnel

The following personnel were supported by this grant:

1. Dr. Michael D. Lemmon, associate professor of electrical engineering, University of Notre Dame
2. Dr. Panos J. Antsakls, full professor of electrical engineering, University of Notre Dame
3. Dr. Peter Szymanski, doctoral student: electrical engineering, University of Notre Dame, grad. date: November 1995.
4. Dr. John Moody, doctoral student, electrical engineering, University of Notre Dame, est. grad. date: Dec. 1997.
5. Mr. Christopher J. Bett, doctoral student, electrical engineering, University of Notre Dame, est. grad. date: May 1998.
6. Mr. Kevin X. He, master student, electrical engineering, University of Notre Dame, est. grad date: May 1998

Report of Inventions

There were no inventions generated by this project.

APPENDIX A

Szymanski, P.T., Lemmon, M.D.,and Bett, C.J. (1996)
"Hybrid Interior Point Training of Modular Neural Networks"
to appear in *Neural Networks*.
Also released as Technical Report of the ISIS Group at the
University of Notre Dame, Report:ISIS-96-007, December 1996.

Hybrid Interior Point Training of Modular Neural Networks

Peter T. Szymanski, Michael Lemmon, and Christopher J. Bett*
Dept. of Electrical Engineering
University of Notre Dame
Notre Dame, IN 46556

August 9, 1997

Abstract

Modular neural networks use a single gating neuron to select the outputs of a collection of agent neurons. Expectation-maximization (EM) algorithms provide one way of training modular neural networks to approximate nonlinear functionals. This paper introduces a hybrid interior-point (HIP) algorithm for training modular networks. The HIP algorithm combines an interior-point linear programming (LP) agorithm with a Newton-Raphson iteration in such a way that the computational efficiency of the interior-point LP methods is preserved. The algorithm is formally proven to converge asypmtotically to locally optimal networks with a total computational cost that scales in a polynomial manner with problem size. Simulation experiments show that the HIP algorithm produces networks whose average approximation error is better than that of EM-trained networks. These results also demonstrate that the computational cost of the HIP algorithm scales at a slower rate than the EM-procedure; and that for small-size networks, the total computational costs of both methods are comparable.

Keyworks: modular neural networks, training, algorithms, interior-point methods, expectation-maximization methods

*The authors would like to acknowledge the partial financial support of the National Science Foundation (MSS92-16559), the Electric Power Research Institute (RP8030-06), and the Army Research Office (DAAH04-95-1-0600, DAAH04-96-0134)

1 Introduction

An artificial neural network is a distributed computing paradigm consisting of a set of self-similar processing units called *neurons*. In many neural network architectures, the network's output, $\hat{y}$, has the form

$$\hat{y}(\bar{z}) = \sum_i f(\bar{\phi}_m^T \bar{z}) \tag{1}$$

where $\bar{\phi}_m$ and $\bar{z}$ are vectors in $\mathbb{R}^R$ and f is a mapping from $\mathbb{R}$ into $\mathbb{R}$. Given a *target mapping*, $y : \mathbb{R}^R \rightarrow R$, we are interested in designing a network whose outputs form an approximation, $\hat{y}$, of the target mapping. Neural networks are well-known to form good approximations of complex input-output mappings. This *universal approximation* [7] [2] [12] ability of neural networks is probably one of the most important reasons for their use. In the networks represented by equation (1), each neuron contributes to the network's output. In certain cases, however, it is reasonable to expect that the structure of the target mapping will be different in disjoint regions of the input space. In such cases, there is little benefit to be gained by additively mixing the outputs of the neurons. It might be better to use the output of a single neuron to exclusively compute the network approximation. Such networks can be viewed as a collection of *experts* where each neuron is an expert at approximating the target function over a region in $\mathbb{R}^R$. These *mixture of expert* or *modular* networks were first introduced in [8].

The training of modular networks can be carried out using *Expectation-Maximization* (EM) procedures [3] [9]. Results in [14] raised concerns about the applicability of EM algorithms due to their linear convergence rate. That work called for the use of more sophisticated optimization methods whose computational costs are well understood. This paper presents just such an algorithm for the training of modular networks using Gaussian activation functions to form piecewise linear approximations. The algorithm is based on a modification of a primal interior point linear programming (LP) algorithm [5] [10]. The resulting algorithm combines an interior-point LP algorithm with a Newton-Raphson (NR) iteration in a way that preserves the computational efficiency of the original interior-point LP algorithm. The resulting algorithm is referred to as the *hybrid interior point* or HIP algorithm. A small-step version of the HIP procedure was first presented in [11]. One principal contribution of this paper is a formal analysis of the HIP procedure's asymptotic behaviour and computational complexity. The second principal contribution of this paper is a detailed simulation which compares the performance of the HIP algorithm and Expectation-Maximization procedure.

The remainder of this paper is organized as follows. Modular neural networks are introduced in

section 2. THe HIP algorithm is presented in section 3. Theoretical results on the HIP algorithm's asymptotic behaviour and computational costs are summarized in section 4. Formal proofs for the results in section 4 will be found in appendix A. Detailed simulation experiments comparing the HIP and EM algorithms will be found in section 5. This section also examines the use of the HIP algorithm in training a modular network to model the input/output behaviour of a fossil fuel electric power generating plant. A final discussion will be found in section 6.

2 Modular Neural Networks

This paper adopts a viewpoint in which neural network training is treated as a function approximation problem [13]. Let $y : \mathbb{R}^R \rightarrow \mathbb{R}$ be a continuous function from $\mathbb{R}^R$ into $\mathbb{R}$ which we call the *target mapping.* It is assumed that y is unknown, except at a finite set of input/output pairs. The set of N input/output pairs is denoted as

$$\mathcal{T} = \{(\bar{z}_1, y(\bar{z}_1)), (\bar{z}_2, y(\bar{z}_2)), \ldots, (\bar{z}_N, y(\bar{z}_N))\} \quad , \tag{2}$$

where $\bar{z}_i \in \mathbb{R}^R$ $(i = 1, \ldots, N)$ is the *input* and $y(\bar{z}_i) \in \mathbb{R}$ is the associated *output.* The set, $\mathcal{T}$, will usually be referred to as the *training set.* The objective of the *training problem* is to find a functional, $\hat{y} : \mathbb{R}^R \rightarrow \mathbb{R}$ from a known parameterized set of functionals, $\mathcal{Y}$, which minimizes the *approximation error's* size over the training set. In this paper, the approximation will be realized by a special class of modular neural network.

The structure of a *modular network* is shown in Figure 1. The network uses a *gating neuron* or *gate* to select between the approximations generated by a collection of *agent neurons* or *agents.* Let $\bar{z} \in \mathbb{R}^R$ be the input to the network. In response to this input, each agent produces an ordered pair of outputs, $(a_m(\bar{z}), \hat{y}_m(\bar{z}))$. $a_m : \mathbb{R}^R \rightarrow \mathbb{R}$ is a functional from the input space, $\mathbb{R}^R$, into the non-negative Reals. The functional measures the *activity level* of the mth agent in response to an input $\bar{z} \in \mathbb{R}^R$. $\hat{y}_m : \mathbb{R}^R \rightarrow \mathbb{R}$ is a functional from the input space into the Reals representing the mth agent's approximation of the target mapping at $\bar{z}$. The gate chooses that agent with the largest activity level and the selected agent's approximation is then used as the network's output. Mathematically, the network's output in response to $\bar{z}$ is written as

$$\hat{y}(\bar{z}) = \hat{y}_m(\bar{z}) \tag{3}$$

$$m = \arg \max_{1 \leq m \leq M} a_m(\bar{z}) \tag{4}$$

Linear Gaussian (LG) modular networks arise when the approximation function is linear and the activation functions are Gaussian. In particular, we assume that the mth agent's activation function has the form,

$$a_m(\bar{z}) = q(m) e^{s\|\bar{z} - \bar{\omega}_m\|^2} \tag{5}$$

where $q(m) \in \mathbb{R}$, $\bar{\omega}_m \in \mathbb{R}^R$, s is a negative real number, and $\sum_m q(m) = 1$. The activation function is parameterized by the scalar $q(m)$ and the R-vector $\bar{\omega}_m$. The mth agent's approximation function has the form

$$\hat{y}_m(\bar{z}) = \bar{\phi}_m^T \bar{z} \tag{6}$$

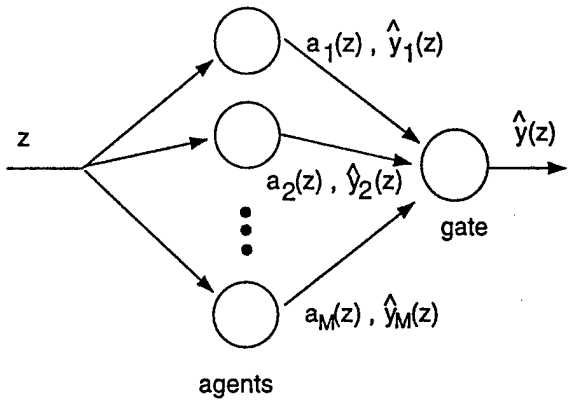


Figure 1: Modular Neural Network

where $\bar{\phi}_m \in \mathbb{R}^R$. This approximation is parameterized by the R-vector $\bar{\phi}_m$. We therefore see that the mth agent is parameterized by the scalar $q(m)$ and the two R-vectors $\bar{\omega}_m$ and $\bar{\phi}_m$. In our later discussion, it will be convenient to aggregate these parameter vectors $\bar{\omega}_m$ and $\bar{\phi}_m$ into a single parameter vector $\bar{\theta}_m \in \mathbb{R}^{2R}$. In particular, we will let $\bar{\theta}_m = \left[\begin{array}{cc} \bar{\phi}_m^T & \bar{\omega}_m^T \end{array} \right]^T$. The vectors,

$$\bar{\Theta} = \left[\begin{array}{cccc} \bar{\theta}_1^T & \bar{\theta}_2^T & \ldots & \bar{\theta}_M^T \end{array} \right]^T \tag{7}$$

$$\bar{q} = \left[\begin{array}{cccc} q(1) & q(2) & \ldots & q(M) \end{array} \right]^T \tag{8}$$

provide a parametrization $(\bar{q}, \bar{\Theta})$ of the entire network.

The squared *output* approximation error over training set $\mathcal{T} = \{(\bar{z}_j, y(\bar{z}_j)\}_{j=1,\ldots,N}$ is denoted as

$$d^2_{\text{out}}(\bar{q}, \bar{\Theta}) = \sum_{j=1}^{N} (y(\bar{z}_j) - \hat{y}(\bar{z}_j))^2 \tag{9}$$

$$= \sum_{j=1}^{N} \sum_{m=1}^{M} Q(m|j)(y(\bar{z}_j) - \bar{\phi}_m^T \bar{z}_j)^2 \tag{10}$$

The second equation arises by expanding $\hat{y}(\bar{z}_j)$ and letting

$$Q(m|j) = \begin{cases} 1 & \text{if } a_m(\bar{z}_j) \geq a_n(\bar{z}_j) \text{ for all } 1 \leq n \leq M \\ 0 & \text{otherwise} \end{cases} \tag{11}$$

The vector

$$\bar{Q} = \left[\begin{array}{ccccccc} Q(1|1) & Q(1|2) & \cdots & Q(m|j) & \cdots & Q(M|N) \end{array} \right]^T \tag{12}$$

of $Q(m|j)$ introduced in equation 11 will be called the network's *selection strategy* since it denotes which agent is associated with which input vector. Note that for a given network parameterization,

$(\bar{q}, \bar{\Theta})$, that there will be a unique selection strategy, $\bar{Q}$. This means that the network can also be described by the ordered pair, $(\bar{Q}, \bar{\Theta})$.

A naive approach to the training problem would involve minimizing $d^2_{\text{out}}(\bar{q}, \bar{\Theta})$ with respect $\bar{q}$ and $\bar{\Theta}$. This problem, however, is ill-posed [6] in the sense that solutions may not be continuous with respect to the data given in the training set. To obtain a well-posed training problem, it is essential that the original output-error criterion be augmented with a regularization term. One common regularization term presents the error of the mth agent on the jth input as

$$d^2_m(j) = \left(y(\bar{\mathbf{z}}_j) - \bar{\phi}_m^T \bar{\mathbf{z}}_j\right)^2 + \|\bar{\omega}_m - \bar{\mathbf{z}}_j\|^2 \tag{13}$$

The total network error then has the form,

$$d^2(\bar{q}, \bar{\Theta}) = \frac{1}{N} \sum_{j=1}^{N} \sum_{m=1}^{M} Q(m|j) d^2_m(j) \tag{14}$$

where $Q(m|j)$ is as defined in equation 11.

The training problem involves minimizing $d^2(\bar{q}, \bar{\Theta})$ with respect to $\bar{q}$ and $\bar{\Theta}$. This is a nonlinear optimization problem which can be very difficult to solve. An alternative (heuristic) approach to solve this problem involves optimizing with respect to the selection strategy. Since a network parametrization $(\bar{q}, \bar{\Theta})$ has an associated selection strategy, $\bar{Q}$, we can attempt to optimize a network with regard to $\bar{\Theta}$ and the selection strategy, $\bar{Q}$. This optimization problem is easier to solve since it is linear in $\bar{Q}$ and quadratic in $\bar{\Theta}$. Note, however, that for a given selection strategy, $\bar{Q}$, there may not be a set of $\bar{q}$ that realizes this strategy. We therefore need to introduce a heuristic way of relating $\bar{q}$ to $\bar{Q}$. One useful heuristic is to view $q(m)$ as the marginal probability of the mth neuron being activated. One estimate of this probability is given by

$$q(m) = \frac{1}{N} \sum_{j} Q(m|j) \tag{15}$$

With the preceding discussion, we can now state the training problem as follows. It is first assumed that there exists a design set consisting of N training points, $\mathcal{T} = \{(y_j, \bar{\mathbf{z}}_j) \in \mathbb{R} \times \mathbb{R}^R, j = 1, \ldots, N\}$ where y_j represents a desired output in response to an input, $\bar{\mathbf{z}}_j$. The training problem is

Problem 1

$$\begin{aligned}
&\textit{minimize:} && \textstyle\sum_{j=1}^{N} \sum_{m=1}^{M} \frac{1}{N} Q(m|j) \left[\|\bar{\mathbf{z}}_j - \bar{\omega}_m\|^2 + (y_j - \bar{\phi}_m^T \bar{\mathbf{z}}_j)^2\right] \\
&\textit{with respect to:} && Q(m|j), \bar{\omega}_m, \bar{\phi}_m (m = 1, \ldots, M; j = 1, \ldots, N) \\
&\textit{subject to:} && Q(m|j) \geq 0 \\
& && \textstyle\sum_{m=1}^{M} Q(m|j) = 1, (j = 1, \ldots, N)
\end{aligned}$$

where $Q(m|j) \in \mathbb{R}$, $\bar{\omega}_m \in \mathbb{R}^R$, and $\bar{\phi}_m \in \mathbb{R}^R$ $(m = 1, \ldots, M; j = 1, \ldots, N)$. The problem is to determine the set of parameters, $\left[\bar{Q}^T, \bar{\Theta}^T\right]^T \in \mathbb{R}^{MN} \times \mathbb{R}^{2MR}$, which minimize the objective where

$$\bar{Q}^T = [Q(1|1), Q(1|2), \ldots, Q(m|j), \ldots] \tag{16}$$

and

$$\bar{\Theta}^T = \left[\theta_1^T, \theta_2^T, \ldots, \theta_M^T\right] \tag{17}$$

with $\theta_m^T = \left[\bar{\phi}_m^T, \bar{\omega}_m^T\right] \in \mathbb{R}^R \times \mathbb{R}^R$. This procedure directly estimates the network parameters $\bar{\Theta}$. The network parameters $\bar{q}$ are then obtained from $\bar{Q}$ using equation 15. Note that this problem is convex with respect to $\bar{Q}$ and $\bar{\Theta}$, separately. The special structure of this problem means that it can be solved by using Expectation-Maximization (EM) algorithms or by using the HIP algorithm presented in the next section.

This section has introduced a set of approximations that can be realized as modular linear Gaussian neural networks. The modular LG networks were characterized by the parameters, $(\bar{q}, \bar{\Theta})$. The regularized training problem was stated with respect to this parameterization. It was noted that solving this problem is more easily accomplished if we optimize the network with respect to the selection strategies, $\bar{Q}$. This observation led to a training problem statement whose performance measure is linear with respect to the selection strategies $\bar{Q}$ and is quadratic with respect to the other network parameters $\bar{\Theta}$. Problems of this form can be solved using expectation-maximization (EM) or hybrid interior-point (HIP) algorithms. The next section introduces the HIP algorithm.

3 Hybrid Interior Point (HIP) Training Algorithm

The training problems under consideration consist of those which may be decomposed into linear and quadratic subproblems. These problems can be written in an alternate form which emphasizes their dependence upon disjoint subsets of their parameters. Letting $\bar{\mathbf{x}} = \bar{Q}$ and letting

$$\bar{\mathbf{c}}(\bar{\Theta}) = \left[\begin{array}{cccc} \frac{1}{N}d_1(1) & \frac{1}{N}d_1(2) & \ldots & \frac{1}{N}d_m(j) \quad \ldots \end{array} \right]^T \tag{18}$$

with $d_m(j) = \|\bar{\mathbf{z}}_j - \bar{\omega}_m\|^2 + (y_j - \bar{\phi}_m^T \bar{\mathbf{z}}_j)^2$, Problem 1 can be rewritten as

Problem 2

$$\begin{array}{rl} \textit{minimize:} & \bar{\mathbf{c}}(\bar{\Theta})^T \bar{\mathbf{x}} \\ \textit{with respect to:} & \bar{\mathbf{x}}, \bar{\Theta} \\ \textit{subject to:} & A\bar{\mathbf{x}} = \bar{\mathbf{b}}, \bar{\mathbf{x}} \geq 0 \end{array}$$

where $A = [I_{N\times N} \ldots I_{N\times N}] \in \mathbb{R}^{N\times MN}$ and $\bar{\mathbf{b}} = [1, \ldots, 1]^T \in \mathbb{R}^N$. The rewritten objective in Problem 2 is linear in the parameters, $\bar{\mathbf{x}}$, and is quadratic in the parameters, $\bar{\Theta}$.

The dependence upon the two distinct parameter sets suggests that an *alternating minimization* (AM) strategy can be employed to solve Problem 1. In an AM procedure, the original optimization problem is decomposed into several subproblems which are iteratively solved by component minimizers for each subproblem. Subsection 3.1 briefly describes the notion of an alternating minimization algorithm. This subsection introduces the hybrid interior-point (HIP) algorithm as an example of an AM procedure. Subsection 3.2 discusses the component minimizers that are used in the HIP algorithm. Subsection 3.3 shows how the two component minimizers can be efficiently combined to produce a computationally efficient algorithm.

3.1 Alternating Minimization

Let $h : \mathbb{R}^R \rightarrow \mathbb{R}$ be a positive semi-definite functional over $\mathbb{R}^R$. Consider the problem of minimizing this functional $h(\bar{\Phi})$ with respect to the parameter vector, $\bar{\Phi}$. It is assumed that the parameter vector can be decomposed as

$$\bar{\Phi}^T = \left[\begin{array}{cccc} \bar{\Phi}_1^T & \bar{\Phi}_2^T & \ldots & \bar{\Phi}_K^T \end{array} \right] \tag{19}$$

where $\bar{\Phi}_i \in \mathbb{R}^{R_i}$ $(i = 1, \ldots, K)$ and $\bar{\Phi} \in \mathbb{R}^R$ where $R = \sum_i R_i$. An *alternating minimization* (AM) procedure attempts to solve this problem by iteratively minimizing the functional with respect to $\bar{\Phi}_i$ for $i = 1, \ldots, K$ while holding the other subvectors, $\bar{\Phi}_j$ $(j \neq i)$, constant. AM procedures can be implemented by two nested loops. The inner loop performs the K component minimizations of

$h(\bar{\Phi})$ with respect to $\bar{\Phi}_i$ $(i = 1, \ldots, K)$ while holding $\bar{\Phi}_j$ constant for $j \neq i$. The outer loop encloses the inner loop and controls the algorithm's iterations until some stopping criterion is satisfied. Algorithm 3.1 gives a pseudo-code description of a general AM algorithm.

Algorithm 3.1 (General AM algorithm)

```
Initialize
   j = 0.
   Choose initial feasible Φ̄(0) = ⟨Φ̄1(0), Φ̄2(0), ..., Φ̄k(0)⟩.
repeat
   for i = 1 to K
      minimize f(Φ̄(j))
      with respect to Φ̄i(j)
      subject to Φ̄k(j) held constant for k ≠ i
   end
   j = j + 1
until(stopping criterion is satisfied)
```

Expectation-Maxmization (EM) algorithms [14] provide well-known examples of AM procedure. The HIP procedure described below and first introduced in [11] provides another example of an AM procedure. The HIP algorithm consists of a parameterized outer loop, whose iteration produces a sequence that converges to a local optimum of the training problem. Each element within that sequence is generated by the inner loop. This inner loop consists of a component linear and quadratic minimizer. A pseudo-code listing for the HIP algorithm is shown in algorithm 3.2 . The behaviour of this algorithm is controlled by parameters β and γ.

The outer loop in the HIP algorithm generates a sequence of solutions $\{\bar{\mathbf{x}}^{(k)}, \bar{\Theta}^{(k)}\}$. Associated with the kth iteration in the loop is a parameter $\alpha^{(k)} \geq 0$ which is a term in a monotone increasing sequence of the form

$$\alpha^{(k+1)} = \beta\alpha^{(k)} \tag{20}$$

where $\beta > 1$. $\alpha^{(k)}$ is a barrier coefficient used by the component linear minimizer. The parameter β is a free parameter of the algorithm controlling the number of iterations in the outer loop of the HIP algorithm.

The inner loops of the HIP algorithm consist of the component linear and quadratic minimizers. The linear minimizer solves the linear programming (LP) problem associated with minimizing the network error with respect to the selection strategies, $\bar{Q}$, subject to $\bar{\Theta}$ being constant. The quadratic minimizer uses a Newton-Raphson (NR) descent method to minimize the network error with respect to $\bar{\Theta}$ subject to the selection strategies, $\bar{Q}$, being constant. This minimizer is controlled by the

parameter γ. The combination of linear IP solvers with Newton-Raphson (NR) descent methods forms the *hybrid* interior point or HIP algorithm. Detailed discussion of each component minimizer will be found in subsection 3.2. Subsection 3.3 shows how to efficienctly combine both minimizers.

Algorithm 3.2 (Hybrid IP AM algorithm)

Initialize
$k = 0$.
Choose $\alpha^{(k)} > 0$, $\bar{\Theta}^{(k)}$, and initial $\bar{\mathbf{x}}^{(k)}$.
repeat
$\alpha^{(k+1)} = \beta\alpha^{(k)}$, $\beta > 1$.
Linear update:
$\bar{\mathbf{x}}_0 = \bar{\mathbf{x}}^{(k)}$
$i = 0$
while$\left(\|P_{AX_i} X_i(\alpha^{(k+1)}\bar{\mathbf{c}}(\bar{\Theta}) - \bar{\mathbf{x}}_i^{-1})\| > 0.2401\right)$
$\bar{\mathbf{x}}_{i+1} = \bar{\mathbf{x}}_i - X_i P_{AX_i} X_i(\alpha^{(k+1)}\bar{\mathbf{c}}(\bar{\Theta}^{(k)}) - \bar{\mathbf{x}}_i^{-1})$
$i = i + 1$
end
$\bar{\mathbf{x}}^{(k+1)} = \bar{\mathbf{x}}_i$
Quadratic update:
For $m = 1, \ldots, M$
$\bar{\theta}^{(k+1)} = (1 - \gamma^{(k)})\bar{\theta}^{(k)} + \gamma^{(k)}\bar{\theta}^{(k+1),*}$
end
$k = k + 1$
until$(\|(\bar{\mathbf{x}}^{(k)}, \bar{\Theta}^{(k)}) - (\bar{\mathbf{x}}^*, \bar{\Theta}^*)\| < \epsilon)$

3.2 Component Minimizers

The HIP algorithm combines an interior-point linear programming algorithm with a quadratic minimization procedure. Both of these component minimizers are discussed in detail below.

Linear Minimizer

The interior-point (IP) method for solving the LP subproblems is a barrier method that uses logarithmic barriers to keep solutions inside the feasible set. The method generates a sequence of points that follow a path in the interior of the polyhedral set of feasible solutions. The resulting central path guides the approximate solutions to the optimal solution on the feasible set's boundary. The path following algorithm is fast, taking $O(\sqrt{n}L)$ iterations to converge with a computational cost of $O(n^{3.5}L)$ floating point operations. n is the dimension of the LP problem; and L denotes the "size" of the LP problem, being the number of bits required to represent the coefficients of A, $\bar{\mathbf{b}}$, and $\bar{\mathbf{c}}$. (Note: in our applications the LP problems will be of size $n = NM$).

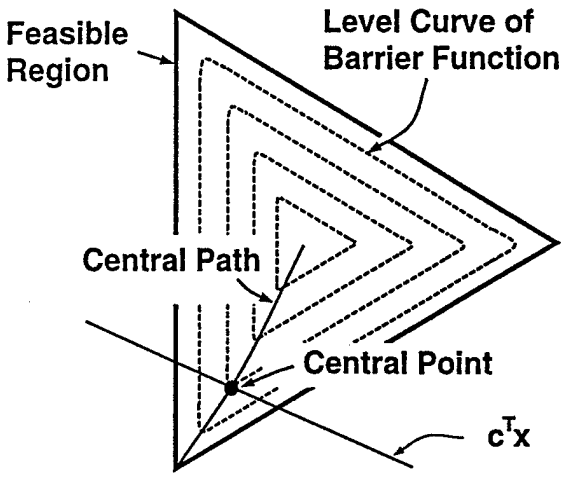


Figure 2: Interior Point concepts

The IP method is a path following algorithm [5] which solves the following sequence of optimization problems. Each problem is based upon a primal parameterization of Problem 2 when minimized solely with respect to $\bar{x}$. Let the kth solution to this problem be denoted as $\bar{x}^{(k)}$ with ith component $x_i^{(k)}$. The kth problem has the form,

$$\begin{aligned} \text{minimize:} \quad & \alpha^{(k)} \bar{c}^T \bar{x}^{(k)} - \textstyle\sum_i \log x_i^{(k)} \\ \text{with respect to:} \quad & \bar{x}^{(k)} \\ \text{subject to:} \quad & A\bar{x}^{(k)} = \bar{b}, \bar{x}^{(k)} \geq 0 \end{aligned} \tag{21}$$

where $\alpha^{(k)} \geq 0$ $(k = 1, \cdots, K)$ is a monotone increasing sequence of real numbers generated by the iteration $\alpha^{(k+1)} = \beta\alpha^{(k)}$ where $\beta > 1$. $\bar{x}^*(\alpha^{(k)})$ denotes the optimal solution for the k^{th} optimization problem in the sequence and is referred to as a *central point.* The locus of all points, $\bar{x}^*(\alpha^{(k)})$ where $\alpha^{(k)} \geq 0$, is called the *central path.* The augmented problem takes the original LP cost function and adds a logarithmic barrier which keeps the central points away from the boundaries of the feasible set. As α increases, the effect of the barrier is decreased, thereby allowing the k^{th} central point to approach the LP problem's optimal solution in a controlled manner. Figure 2 depicts the feasible set for a sample LP problem along with the central path, central points, and a set of level curves corresponding to contours of the barrier function.

Interior point algorithms solve the LP problem by approximately solving the sequence of augmented problems shown in (21). The approximate central points are computed using one or more scaling steepest descent (SSD) updates of the form

$$\bar{x}^{(k+1)} = \bar{x}^{(k)} - XP_{AX}X(\alpha^{(k+1)}\mathbf{c} - (\bar{x}^{(k)})^{-1}) \tag{22}$$

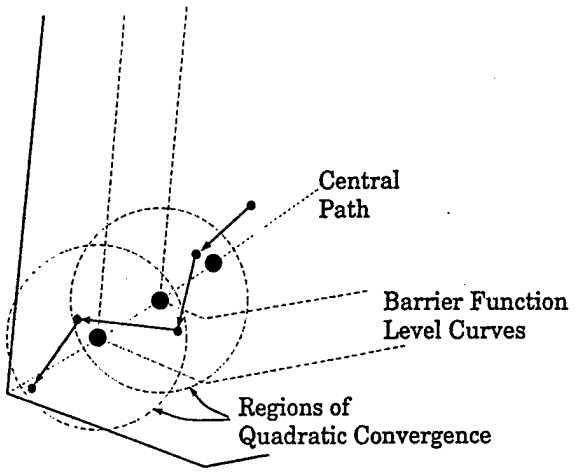


Figure 3: Path Following Example

where $X = \text{diag}(x_1, x_2, \ldots, x_n) \in \mathbb{R}^{n \times n}$ is a scaling transformation, $P_A = I - A^T(AA^T)^{-1}A \in \mathbb{R}^{n \times n}$ is an orthogonal projection, and $\bar{\mathbf{x}}^{-1} = \left[x_1^{-1}, x_2^{-1}, \ldots, x_n^{-1}\right]^T$. Figure 3 illustrates this sequence of approximate central points.

The interior-point algorithm can be proven to converge to an ϵ-neighborhood of the optimal solution to the LP problem provided these approximate solutions are close to the central path. The following *proximity measure* is essential for characterizing this notion of "nearness" to the central path,

$$\delta(\bar{\mathbf{x}}^{(k)}, \alpha^{(k)}) = ||P_{AX^{(k)}} X^{(k)}(\alpha^{(k)}\bar{\mathbf{c}} - (\bar{\mathbf{x}}^{(k)})^{-1})|| \tag{23}$$

$\delta(\bar{\mathbf{x}}^{(k)}, \alpha^{(k)}) < 1$ places $\bar{\mathbf{x}}^{(k)}$ in the region of quadratic convergence for the SSD algorithm. Once a point is near the central path and satisfies the *proximity condition*,

$$\delta(\bar{\mathbf{x}}^{(k)}, \alpha^{(k)}) \leq 0.5 \ , \tag{24}$$

the IP algorithm must maintains the proximity of successive points to the central path. This is generally done by selecting a β and then using a sufficient number of SSD steps to enforce the proximity condition. If β is chosen so that only a single SSD step is required, then the IP algorithm is said to be a *small-step* procedure. Small-step algorithms have been shown to converge in $O(\sqrt{n}L)$ iterations with an associated computational cost of $O(n^{3.5}L)$ [5]. If a larger value of β is used, then several SSD steps may be required to enforce the proximity condition. Such algorithms are often referred to as *large-step* IP algorithms. In general, large-step algorithms can exhibit considerably lower total computational cost than their small-step counterparts.

Quadratic Minimizer

The quadratic minimizer minimizes $\bar{\mathbf{c}}(\bar{\Theta})^T\bar{\mathbf{x}}$ with respect to $\bar{\Theta}$. The determination of successive quadratic parameters is accomplished using standard descent techniques to optimize $\bar{\mathbf{c}}(\bar{\Theta})^T\bar{\mathbf{x}}$ with respect to $\bar{\Theta}$. The determination of successive quadratic parameter estimates is termed the Θ-update.

The Θ-update uses a Newton-Raphson (NR) approach to solve the quadratic optimization with respect to $\bar{\Theta}$. The basic form of an NR update is

$$\bar{\Theta}^{(k+1)} = \bar{\Theta}^{(k)} - \left[H(\bar{\Theta}^{(k)})\right]^{-1} \nabla_{\bar{\Theta}}(\bar{\Theta}^{(k)})$$

where $H(\bar{\Theta})$ is the Hessian of the objective functional with respect to $\bar{\Theta}$ and $\nabla_{\bar{\Theta}}(\bar{\Theta})$ is the gradient with respect to $\bar{\Theta}$ evaluated at point $\bar{\Theta}^{(k)}$. Typically, the update is performed iteratively for general problems. The optimization is quadratic, however, so the following closed form solution results: $\bar{\Theta}^* = [\bar{\theta}_1^*, \bar{\theta}_2^*, \ldots, \bar{\theta}_M^*]$ where $\bar{\theta}_m^* = [\bar{\phi}_m^*, \bar{\omega}_m^*]$,

$$\bar{\phi}_m^* = \left[\sum_{j=1}^{N} \frac{1}{N} Q(m|j)\bar{\mathbf{z}}_j\bar{\mathbf{z}}_j^T\right]^{-1} \sum_{j=1}^{N} \frac{1}{N} Q(m|j)\bar{\mathbf{z}}_j y_j \tag{25}$$

and

$$\bar{\omega}_m^* = \frac{\sum_{j=1}^{N} \frac{1}{N} Q(m|j)\bar{\mathbf{z}}_j}{\sum_{j=1}^{N} \frac{1}{N} Q(m|j)}. \tag{26}$$

These minimizing solutions assume that $\sum_{j=1}^{N} \frac{1}{N} Q(m|j) \neq 0$ and that $\left[\sum_{j=1}^{N} \frac{1}{N} Q(m|j)\bar{\mathbf{z}}_j\bar{\mathbf{z}}_j^T\right]$ is nonsingular. The first condition will always be satisfied as at least one sample point will always be assigned to a model ($\forall m, \exists j$ such that $Q(m|j) > 0$). The second condition can be violated for insufficient amounts of data or when the data is linearly dependent. The problem can be corrected in such a case by reducing the number of models or increasing the number of samples to ensure the linear independence of the data.

3.3 Combining Component Minimizers

The behaviour of the HIP algorithm is controlled by the parameters, β and γ. Choices for β are dictated by the desired number of iterations in the outer loop of the HIP algorithm. If a small enough β is used, then we have a *small-step* HIP algorithm. For larger β, a *large-step* HIP procedure results. The primary difference between these methods lies in the number of SSD iterations required in the linear component minimzer. β is therefore chosen to control the total number of iterations in the outer loop of the HIP algorithm. The effect of specific β choices on the algorithm's total computational cost is discussed in section 4. These results suggest that choices

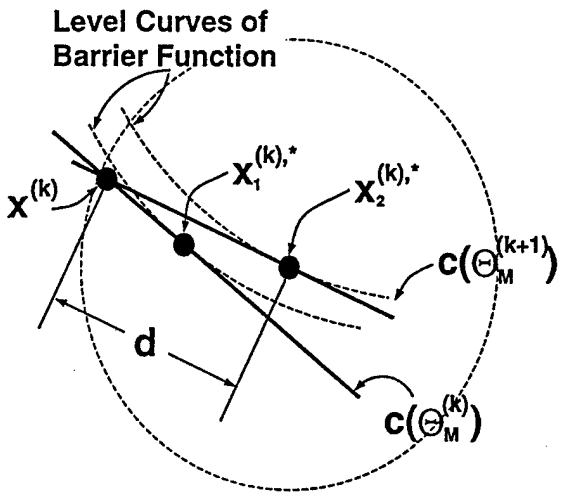


Figure 4: Effect of cost plane rotation

for β resulting in large-step HIP algorithms should be fastest. The simulation results in section 5 supports this assertion. Choices for γ are important for ensuring that the component minimzers work well together. The following subsection summarizes the desirable choices for γ. Formal proofs of justifying the desirability of these choices will be found in section 4 and appendix A.

A key concern in the HIP algorithm is the proximity of intermediate solutions in the sequence, $\{\bar{\mathbf{x}}^{(k)}, \bar{\Theta}^{(k)}\}$, to the central path. The proximity of successive solutions to the central path (see eq. (24)) results in the attractive convergence properties for the basic IP algorithm. This condition must be maintained by the HIP algorithm if the attractive computational properties of the interior-point method is to be preserved. Note, however, that performing the quadratic minimization changes the parameter vector, $\bar{\Theta}$, and hence changes the cost vector, $\bar{\mathbf{c}}(\bar{\Theta})$ (See Figure 4). It is possible that changes to the cost vector will rotate the cost vector about the current solution $\bar{\mathbf{x}}^{(k)}$ in such a way that $\bar{\mathbf{x}}^{(k)}$ no longer satisfies the proximity condition in (24). To make sure that the proximity condition is not violated, it is necessary to constrain the step size of the $\bar{\Theta}$-update by choosing the γ parameter appropriately. In section 4 and appendix A it is shown that the appropriate choice is

$$\gamma^{(k)} < \frac{0.12995}{n\|\bar{\mathbf{c}}(\bar{\Theta}^{(k+1),*}) - \bar{\mathbf{c}}(\bar{\Theta}^{(k)})\|}. \tag{27}$$

For this choice of γ, the HIP algorithm preserves the computational efficiency of the underyling interior-point linear programming algorithm.

4 Theoretical results

The utility of HIP algorithms rests in their convergence and computational properties. This section summarizes the key technical results supporting the algorithm's asserted computational efficiency. The results pertain to small-step versions of the HIP algorithm, but related results to the large-step version can be obtained using similar arguments. Proofs for the following results appear in [15], [5], and in appendix A. The results deal with proximity, convergence, and computational properties.

4.1 Proximity

As discussed earlier, the HIP algorithm must maintain the proximity condition (24). This proximity condition is enforced by using the appropriate number of SSD iterations for a given β. The following classical results [5] reparameterize β with respect to ν using the following equation,

$$\beta = 1 + \nu/\sqrt{n} \tag{28}$$

When $\nu \in (0, 0.1]$, then we say the algorithm is a *small-step* procedure. Theorem 1 and Corollary 2 specify how SSD steps affect proximity for a small-step HIP algorithm. Theorem 1 indicates that only a single SSD step is required to maintain proximity in moving from $\bar{\mathbf{x}}^{(k)}$ to $\bar{\mathbf{x}}^{(k+1)}$, while Corollary 2 details the "extra" proximity which is gained by taking multiple SSD steps. Theorem 3 specifies the effect of SSD steps on proximity for large-step HIP algorithms. It states that a linear number of steps ($O(n\mu^2/(1+\mu^2))$ steps with $\mu = \nu/\sqrt{n}$) is required to find $\bar{\mathbf{x}}^{(k+1)}$ satisfying (24) given an initial $\bar{\mathbf{x}}^{(k)}$ which satisfies (24).

Theorem 1 ([5]) *Let* $\delta(\bar{\mathbf{x}}, \alpha, \bar{\Theta}) = \|P_{AX} X(\alpha \bar{\mathbf{c}}(\bar{\Theta}) - \bar{\mathbf{x}}^{-1})\|$ *be the proximity measure where* $\bar{\mathbf{x}}^{-1} = \langle x_1^{-1}, x_2^{-1}, \ldots, x_n^{-1} \rangle^T$ *and* $\alpha > 0$. *If* $\delta(\bar{\mathbf{x}}^{(k)}, \alpha^{(k)}, \bar{\Theta}^{(k)}) \leq 0.5$ *and* $\alpha^{(k+1)} = \alpha^{(k)}(1 + \nu/\sqrt{n})$ *where* $\nu \in (0, 0.1]$, *then one SSD step finds* $(\bar{\mathbf{x}}^{(k+1)}, \bar{\Theta}^{(k)})$, *such that* $\delta(\bar{\mathbf{x}}^{(k+1)}, \alpha^{(k+1)}, \bar{\Theta}^{(k)}) \leq 0.5$.

Corollary 2 (Appendix A) *Assuming the same conditions as in Theorem 1, then J SSD steps produce a point* $(\bar{\mathbf{x}}_J^{(k+1)}, \bar{\Theta}^{(k)})$ *such that* $\delta(\bar{\mathbf{x}}_J^{(k+1)}, \alpha^{(k+1)}, \bar{\Theta}^{(k)}) \leq \delta_1 = (0.7)^{2^J}$.

Theorem 3 ([5]) *Let* $\delta(\bar{\mathbf{x}}, \alpha, \bar{\Theta})$ *be the proximity measure where* $\alpha > 0$. *If* $\delta(\bar{\mathbf{x}}^{(k)}, \alpha^{(k)}, \bar{\Theta}^{(k)}) \leq 0.5$ *and* $\alpha^{(k+1)} = \alpha^{(k)}(1 + \mu)$ *where* $\mu \geq 0.1/\sqrt{n}$, *then* $O(n\mu^2/(1+\mu^2))$ *SSD updates find a new point,* $(\mathbf{x}^{(k+1)}, \bar{\Theta}^{(k)})$, *such that* $\delta(\bar{\mathbf{x}}^{(k+1)}, \alpha^{(k+1)}, \bar{\Theta}^{(k)}) \leq 0.5$.

The preceding results are used in determining the choice of γ introduced in section 3, equation (27). Theorem 4 describes a bound on $\gamma^{(k)}$ that maintains the proximity condition. Note that this bound applies to either the small-step or large-step HIP algorithms.

Theorem 4 (Appendix A) *Let $\bar{\Theta}^{(k)}$ and $\bar{\Theta}^{(k),*}$ be the current and minimizing parameter vectors at time k. Let $\bar{c}^{(k)} = \bar{c}(\bar{\Theta}^{(k)})$ and $\bar{c}^{(k),*} = \bar{c}(\bar{\Theta}^{(k),*})$ and assume $\bar{c}(\bar{\Theta})$ is a non-negative, convex function of $\bar{\Theta}$. Let $\delta(\bar{x}, \alpha, \bar{\Theta})$ be the proximity measure. Assume $\delta(\bar{x}^{(k+1)}, \alpha^{(k+1)}, \bar{\Theta}^{(k)}) = \delta_1 < 0.5$ and let $\bar{\Theta}^{(k+1)} = (1 - \gamma^{(k)})\bar{\Theta}^{(k)} + \gamma^{(k)}\bar{\Theta}^{(k+1),*}$. If $\gamma^{(k)}$ is chosen as*

$$\gamma^{(k)} < \frac{\delta_2 - \delta_1}{2n\|\bar{c}^{(k+1),*} - \bar{c}^{(k)}\|} \tag{29}$$

where $\delta_1 < \delta_2 = 0.5$, then $\delta(\bar{x}^{(k+1)}, \alpha^{(k+1)}, \bar{\Theta}^{(k+1)}) < \delta_2 = 0.5$.

The component minimizers together form an iteration that computes $(\bar{x}^{(k+1)}, \bar{\Theta}^{(k+1)})$ in proximity to $\bar{x}^*(\alpha^{(k+1)}, \bar{\Theta}^{(k+1)})$ given $(\bar{x}^{(k)}, \bar{\Theta}^{(k)})$ which is already in proximity to $\bar{x}^*(\alpha^{(k)}, \bar{\Theta}^{(k)})$. Theorem 5 quantifies the result for both the small-step and large-step cases that is found in equation (27).

Theorem 5 (Appendix A) *Let $\delta(\bar{x}^{(k)}, \alpha^{(k)}, \bar{\Theta}^{(k)}) \leq 0.5$ and let $\alpha^{(k+1)} = \alpha^{(k)}(1 + \mu)$ where $\mu < 0.1/\sqrt{n}$ for the small-step algorithm and $\mu \geq 0.1/\sqrt{n}$ for the large-step case. Let $\bar{\Theta}^{(k+1)} = (1 - \gamma^{(k)})\bar{\Theta}^{(k)} + \gamma^{(k)}\bar{\Theta}^{(k+1),*}$ where*

$$\gamma^{(k)} < \frac{0.12995}{n\|\bar{c}^{(k+1),*} - \bar{c}^{(k)}\|}.$$

Then, one iteration of the small-step variant of Algorithm 3.2 with two SSD steps or one iteration of the large-step variant with $O(n\mu^2/(1+\mu^2))$ SSD steps produces $(\bar{x}^{(k+1)}, \bar{\Theta}^{(k+1)})$ from $(\bar{x}^{(k)}, \bar{\Theta}^{(k)})$ such that $\delta(\bar{x}^{(k+1)}, \alpha^{(k+1)}, \bar{\Theta}^{(k+1)}) \leq 0.5$.

4.2 Asymptotic convergence

The component minimizers' combination as an hybrid IP algorithm is only useful if it converges to an optimum for Problem 1. The preceding theorems only imply that the HIP algorithm will converge to a fixed point, $(\bar{x}^*, \bar{\Theta}^*)$, as $\alpha^{(k)}$ increases. This point corresponds to an optimum of the LP subproblem and an optimum for the quadratic subproblem. Proximity and central paths do not, however, guarantee that $(\bar{x}^*, \bar{\Theta}^*)$ is a local optimum for Problem 1. This section summarizes results establishing the aymptotic convergence of the small-step HIP algorithm to a local optimum. These results first quantify the changes in the linear and quadratic solutions as functions of the LP subproblem's duality gap. Those results then imply that the solution sequence is Cauchy and converges to a fixed point. It is then shown that this fixed point is a local optimum for the regularized training problem.

Convergence results for the HIP algorithm depend heavily on the LP subproblem's duality gap, as do similar results for the LP solvers. The duality gap, $\Delta^{(k)}$, is a non-negative quantity that

denotes an upper bound on the distance to an optimal solution to the LP subproblem in terms of problem cost. An expression for the bound on the duality gap ,$\delta(\bar{\mathbf{x}}, \alpha)$, will be found in [5]. This expression extends to the HIP algorithm where $\delta(\bar{\mathbf{x}}, \alpha)$ becomes $\delta(\bar{\mathbf{x}}, \alpha, \bar{\Theta})$. With this extension, a bound on the duality gap at the k^{th} iteration for the HIP procedure is

$$\Delta^{(k)} \leq \frac{n + \delta(\bar{\mathbf{x}}^{(k)}, \alpha^{(k)}, \bar{\Theta}^{(k)})\sqrt{n}}{\alpha^{(k)}}. \tag{30}$$

$\Delta^{(k)}$ is a function of $\alpha^{(k)}$ since $\delta(\bar{\mathbf{x}}^{(k)}, \alpha^{(k)}, \bar{\Theta}^{(k)})$ can be bounded by 0.5 from the previous proximity conditions. Thus, as $\alpha^{(k)}$ increases, the distance to an optimal LP solution decreases. This property is used in the succeeding results where $\Delta^{(k)}$ takes on a role similar to that of a descent function.

The component linear and quadratic minimizers make incremental changes to $\bar{\mathbf{x}}^{(k)}$ (or $\bar{Q}$) and $\bar{\Theta}^{(k)}$ where the changes can be bounded by a function of the duality gap. These changes rely on satisfaction of proximity conditions and on the forms of the updates, themselves. The changes in $\bar{\mathbf{x}}^{(k)}$ can be related directly to the duality gap. Theorem 6 describes the change in $\bar{\mathbf{x}}$ and provides a bound on $\|\bar{\mathbf{x}}^{(k+1)} - \bar{\mathbf{x}}^{(k)}\|$ in terms of the duality gap. The changes in $\bar{\Theta}$ depend directly on the changes in $\bar{\mathbf{x}}$ due to the updates in (25) and (26). Theorem 7 details the changes in $\bar{\Theta}^*$ and provides a bound on $\|\bar{\Theta}^{(k+1),*} - \bar{\Theta}^{(k),*}\|$ in terms of $\|\bar{\mathbf{x}}^{(k+1)} - \bar{\mathbf{x}}^{(k)}\|$. Both norms have upper bounds which are functions of the duality gap. Both are therefore strictly decreasing as the duality gap is strictly decreasing, implying that $\|\bar{\mathbf{x}}^{(k+1)} - \bar{\mathbf{x}}^{(k)}\| \to 0$ and $\|\bar{\Theta}^{(k+1),*} - \bar{\Theta}^{(k),*}\| \to 0$ as $\Delta^{(k)} \to 0$.

Theorem 6 (Appendix A) *Let $\bar{Q}^{(k)}$ and $\Delta^{(k)}$ be the set of linear parameters (probabilities) and the duality gap at iteration k, respectively. Let $\bar{\mathbf{c}}^{(k)} = \bar{\mathbf{c}}(\bar{\Theta}^{(k)})$ and $\delta(\bar{\mathbf{x}}, \alpha, \bar{\Theta})$ be the proximity measure where $\bar{\mathbf{x}} = \bar{Q}$. Let $\alpha^{(k+1)} = \alpha^{(k)}(1 + \nu/\sqrt{n})$ where $\nu \in (0, 0.1]$. Assume that $\bar{\mathbf{x}}^{(k)}$ satisfies the proximity condition $\delta(\bar{\mathbf{x}}^{(k)}, \alpha^{(k)}, \bar{\Theta}^{(k)}) \leq 0.5$. For two SSD updates, the change in $\bar{Q}$, $\|\Delta Q\| = \|Q^{(k+1)} - Q^{(k)}\|$ is*

$$\|\Delta Q\| < 2(\delta(1 + \nu) + \nu)\Delta^{(k)}. \tag{31}$$

where $\delta = \delta(\bar{\mathbf{x}}^{(k)}, \alpha^{(k)}, \bar{\Theta}^{(k)})$.

Theorem 7 (Appendix A) *Let $Q(m|j) = Q^{(k)}(m|j)$ and $\Delta Q(m|j) = Q^{(k+1)}(m|j) - Q^{(k)}(m|j)$. Define $\bar{\theta}^*_m = \left[(\bar{\phi}^*_m)^T, (\bar{\omega}^*_m)^T\right]^T \in \mathbb{R}^{2R}$ where $\bar{\phi}^*_m$ and $\bar{\omega}^*_m$ are defined as in (25) and (26). Let Q_{CP} be the set of all Q's on or near the central path. Define*

$$\begin{aligned} A_m &= \textstyle\sum_{j=1}^{N} \frac{1}{N} Q(m|j)\bar{\mathbf{z}}_j\bar{\mathbf{z}}_j^T \\ E_m &= \textstyle\sum_{j=1}^{N} \frac{1}{N} \Delta Q(m|j)\bar{\mathbf{z}}_j\bar{\mathbf{z}}_j^T \end{aligned}$$

where $A_m, E_m \in \mathbb{R}^{R\times R}$. *Assume bounded inputs and outputs for the design set* $\mathcal{T}$, $\|\bar{z}\| \leq \zeta$, $|y| \leq Y_{max}$. *Assume that* A_m *is of full rank for* $Q \in Q_{CP}$. *Let* $\mu_{max} = \sup_{Q\in Q_{CP}} \|A_m^{-1}\|$, $\|E_m\| < 1/\mu_{max}$, *and* $\beta_{max} = \sup_{Q\in Q_{CP}} (\sum_{j=1}^{N} \frac{1}{N} Q(m|j))^{-1}$. *Then,*

$$\|\bar{\Theta}^{(k+1),*} - \bar{\Theta}^{(k),*}\| \leq M\mathcal{K}\|\Delta Q\| \tag{32}$$

where $\mathcal{K} = \mu_{max}\zeta Y_{max}(1 + \mu_{max}\zeta^2/(1-r_1)) + \beta_{max}\zeta(1 + \beta_{max}/(1-r_2))$, $r_1 = \|A_m^{-1}\|\,\|E_m\| < 1$, *and* $r_2 = (\sum_{j=1}^{N} \frac{1}{N}\Delta Q(m|j))/(\sum_{j=1}^{N} \frac{1}{N} Q(m|j)) < 1$.

The significance of Theorems 6 and 7 is that the norms of the changes in the parameters have bounds which are functions of the duality gap. The results indicate that the changes in the parameters are guaranteed to decrease as the duality gap decreases. It is initially surprising that both parameter changes can be bounded by the duality gap. However, considering that the problem is expressed as a linear subproblem, it is not surprising that the linear parameters' changes are functions of $\Delta^{(k)}$, which is intrinsic to LP problems. It is also not surprising that the quadratic parameters' changes depend upon $\Delta^{(k)}$ as the closed form solutions for the minimizing parameter vectors depend directly upon the linear parameters which themselves depend upon $\Delta^{(k)}$.

Asymptotic convergence of the small-step variant requires that $\{(\bar{x}^{(k)}, \bar{\Theta}^{(k)})\} \rightarrow (\bar{x}^*, \bar{\Theta}^*)$ as k increases. The preceding two results do not imply this condition. Rather, they imply that the linear parameters converge and the *optimal estimates* of the quadratic parameters converge. In order to prove convergence, both the linear and quadratic parameters must converge simultaneously. For this to occur, the convex combination coefficient, $\gamma^{(k)}$, must tend to and remain fixed at unity as k increases. This implies that the quadratic parameters are always updated exactly as their optimal estimates and that $\|\bar{\Theta}^{(k+1)} - \bar{\Theta}^{(k)}\| = \|\bar{\Theta}^{(k+1),*} - \bar{\Theta}^{(k),*}\|$. When this condition holds, then both $\|\bar{x}^{(k+1)} - \bar{x}^{(k)}\| \rightarrow 0$ and $\|\bar{\Theta}^{(k+1)} - \bar{\Theta}^{(k)}\| \rightarrow 0$. Theorem 8 states that there is an iteration number, K, for which all iterations greater than K satisfy the condition on $\gamma^{(k)}$. Theorem 9 provides the result that $\{(\bar{x}^{(k)}, \bar{\Theta}^{(k)})\} \rightarrow (\bar{x}^*, \bar{\Theta}^*)$. The proof of Theorem 9 shows that the sequence is Cauchy and converges to a fixed point.

Theorem 8 (see Appendix A) *For the HIP algorithm described above, there exists a* $K \geq 0$ *such that* $\gamma^{(k)} = 1$ *for all* $k \geq K$.

Theorem 9 (Appendix A) *For the HIP algorithm described above, if* $\gamma^{(k)} = 1$ *for all* $k \geq K \geq 0$, *then the sequence of solutions,* $\{(\bar{x}^{(k)}, \bar{\Theta}^{(k)})\}$, *converges to a fixed point,* $(\bar{x}^*, \bar{\Theta}^*)$.

The optimality of the fixed point determined by the HIP algorithm is established in the following theorem 10.

Theorem 10 (Appendix A) *Let* $\bar{\mathbf{w}}^* = \left[(\bar{\mathbf{x}}^*)^T, (\bar{\Theta}^*)^T\right]^T$ *be the limit of the sequence,* $\{\bar{\mathbf{w}}^{(k)}\} = \{\left[(\bar{\mathbf{x}}^{(k)})^T, (\bar{\Theta}^{(k)})^T\right]^T\}$, *produced by the HIP algorithm. Let the minimization problem be*

$$\begin{aligned} &\textit{minimize:} && \bar{\mathbf{c}}(\bar{\Theta})^T\bar{\mathbf{x}} \\ &\textit{with respect to:} && \bar{\mathbf{x}}, \bar{\Theta} \\ &\textit{subject to:} && A\bar{\mathbf{x}} = \mathbf{b}, \bar{\mathbf{x}} \geq 0. \end{aligned} \tag{33}$$

$\bar{\mathbf{w}}^* = \left[(\bar{\mathbf{x}}^*)^T, (\bar{\Theta}^*)^T\right]^T$ *is a locally minimum solution to (33).*

4.3 Computational properties

The final two results refer to the small-step variant's computational properties. Small-step HIP variants are direct extensions of small-step interior point LP solvers. The latter algorithms enjoy excellent computational properties, $O(\sqrt{n}\log_2(1/\epsilon))$ iterations convergence rate and $O(n^{3.5}\log_2(1/\epsilon))$ flops cost, to converge to an ϵ-neighborhood of an optimal solution. Under appropriate problem scaling assumptions, the HIP procedure inherits the computational efficiency of its linear counterpart. The extension requires $O(\sqrt{n}\log_2(\sqrt{n}/\epsilon))$ iterations to converge with an associated computational cost of $O((n^{3.5} + n^{1.5}R^2 + \sqrt{n}MR^3)\log_2(\sqrt{n}/\epsilon))$ flops to converge to an ϵ-neighborhood of locally optimal solutions. In this case $n = NM$. The two results are summarized below. Together, they indicate that the small-step variant is computationally efficient and that it will scale well as problem size increases.

Theorem 11 (Appendix A) *Let* $\bar{\mathbf{w}}^{(k)} = \left[(\bar{\mathbf{x}}^{(k)})^T, (\bar{\Theta}^{(k)})^T\right]^T$ *be the* k^{th} *approximate solution generated by the HIP algorithm. Assume that* $\Delta^{(0)} < 1/\epsilon$. *Assume that the cost vector is scaled by* S *such that* $\gamma^{(k)} = 1$ *for all* $k \geq K$ *and that* $2(\delta(1+\nu)+\nu)(1+MK)\Delta^{(K)} > \epsilon$. *Given these assumptions, the HIP algorithm will converge to an* ϵ*-neighborhood of a locally optimal solution* $(\|\bar{\mathbf{w}}^{(k)} - \bar{\mathbf{w}}^*\| < \epsilon)$ *in* $O(\sqrt{n}\log_2(\sqrt{n}/\epsilon))$ *iterations.*

The HIP algorithm's overall computational cost is computed by determining the computational complexity of the individual steps and then multiplying by the algorithm's convergence rate. The linear update executes two SSD steps each employing a matrix inversion with a worst case complexity of $O(n^3)$ where $n = MN$. The quadratic update requires NR^2 multiplications resulting from the vector outer product and employs a matrix inversion with a computational complexity of

$O(R^3)$. Also, the quadratic update is done on M models so the quadratic update has a computational complexity of $O(nR^2)$ or $O(MR^3)$. These complexity estimates multiplied by the estimate of the number of iterations establish the following theorem:

Theorem 12 (Appendix A) *The computational cost of the small-step, HIP algorithm is* $O((n^{3.5} + n^{1.5}R^2 + \sqrt{n}MR^3)\log_2(\sqrt{n}/\epsilon))$

5 Examples

Two collections of experiments examine the hybrid IP algorithms' computational properties and use. The first set performs comparisons between the two hybrid IP algorithms and the EM algorithm. Comparisons in Section 5.1 address the relative computational costs and error characteristics of the three algorithms. The second set of experiments demonstrates the two IP algorithms' use in training networks to determine multiple, linear, set point models of a nonlinear plant. Those experiments use data from a fossil fuel burning, electric power generating plant.

5.1 Mixture density parameter estimation

The first collection of experiments compares the IP algorithms with the EM algorithm when used on mixture density parameter estimation problems. The three algorithms identify the means of a set of normal densities using sample vectors generated from the densities. The densities have means which correspond to the vertices of a simplex in an R dimensional space. Similar to experiments in [14], the means are equispaced. These experiments assume $M = 2$ to 15 candidate densities. Samples sets ranging in size from $N = 300$ to $N = 750$ vectors are the training sets used. All the experiments assume known and equal covariances, $\Sigma = I$.

The specific EM algorithm used in these experiments is a two step procedure modeled after the EM algorithms described in [14] for solving mixture problems. It consists of implementing equations (4.0) and (4.8) from [14]. These equations are

$$q(m)^{(k+1)} = \frac{1}{N} \sum_{j=1}^{N} \frac{q(m)^{(k)} P(\bar{z}_j | m, \bar{\Theta}^{(k)})}{P(\bar{z}_j | \bar{\Theta}^{(k)})} \tag{34}$$

and

$$\bar{\omega}_m^{(k+1)} = \left[\sum_{j=1}^{N} \bar{z}_j \frac{q(m)^{(k+1)} P(\bar{z}_j | m, \bar{\Theta}^{(k)})}{P(\bar{z}_j | \bar{\Theta}^{(k)})} \right] \left[\sum_{j=1}^{N} \frac{q(m)^{(k+1)} P(\bar{z}_j | m, \bar{\Theta}^{(k)})}{P(\bar{z}_j | \bar{\Theta}^{(k)})} \right]^{-1} \tag{35}$$

for $m = 1, \ldots, M$, where

$$P(\bar{z}_j | \bar{\Theta}^{(k)}) = \sum_{m=1}^{M} q(m) k_1 \exp\left(-\kappa_1 \| \bar{z}_j - \bar{\omega}_m \|^2 \right)$$

with k_1 normalizing constant and $q(m)$ being the marginal probability of observing a sample from the m^{th} class. The following experiments assume a value of unity for κ_1 in the exponential in the preceding equation. Different distances between the mixtures' means in the experiments help demonstrate the algorithms' convergence rates and costs for different, relative variances.

The experiments use the EM algorithm above for comparison with the HIP algorithms. The EM algorithm solves a soft clustering problem, while the HIP algorithms solve a hard clustering problem. k-means clustering, a hard version of EM, may be used to solve the problem. The comparisons use the EM algorithm, however, since its implementation resembles the HIP algorithms.

The first set of experiments makes a qualitative analysis of the three methods. This set assumed problem parameter values of $R = 9$, $N = 300$, $M = 2$ to 9, and a spacing of four units between the means. This spacing between the means corresponds to a well separated set of densities whose parameters should be easy to identify. The algorithms were initialized with random initial conditions. In the runs, the algorithms executed until the solutions were within 10^{-5} of an optimal solution. Results from this set examine the convergence rate in iterations and the computational cost in flops.

Results for the first set of experiments appear in Figures 5 and 6. These results serve as a means to rank the algorithms' performance relative to one another. Mean empirical iteration results, along with a "best-fit" characteristic, in Figure 5 exhibit $O(n^{1.53})$, $O(n^{0.45})$, and constant iteration rates for the EM, small-step, and large-step algorithms, respectively, where n is the dimension of the linear subproblem. Mean computational cost results in Figure 6 exhibit $O(n^{2.42})$, $O(n^{1.78})$, and $O(n^{1.53})$ flops cost rate for the EM, small-step, and large-step algorithms, respectively. While the IP variants exhibit lower convergence rates and cost rates than does EM, the plots demonstrate that the EM algorithm incurs lower cost for these well separated problems. As the number of models increases, however, EM performance begins to approach the large-step algorithm's performance in terms of cost. The same cannot be said with respect to the small-step IP variant, which still has a much greater cost. These results allow us to eliminate the small-step variant from further consideration in our comparisons, as it always has much greater cost.

The next sets of experiments perform a more quantitative analysis of the large-step IP variant and the EM algorithm. These experiments assume $R = 19$, $N = 300, 500, 750$ samples, $M = 2$ to 15 mixtures, and a spacing of two units between the densities' means. This spacing corresponds to more closely spaced, harder to identify means. The algorithms terminate in these runs when the solutions are within 5×10^{-5} of an optimal solution.

Computational cost results for the three cases appear in Figures 7-9. These graphs display the mean computational costs for the two algorithms along with bounds indicating the maximum and minimum costs incurred for the various sized mixtures. The cost rates for the EM and large-step algorithms are $O(n^{1.7})$ and $O(n^{1.45})$ flops, respectively. The two algorithms are comparable,

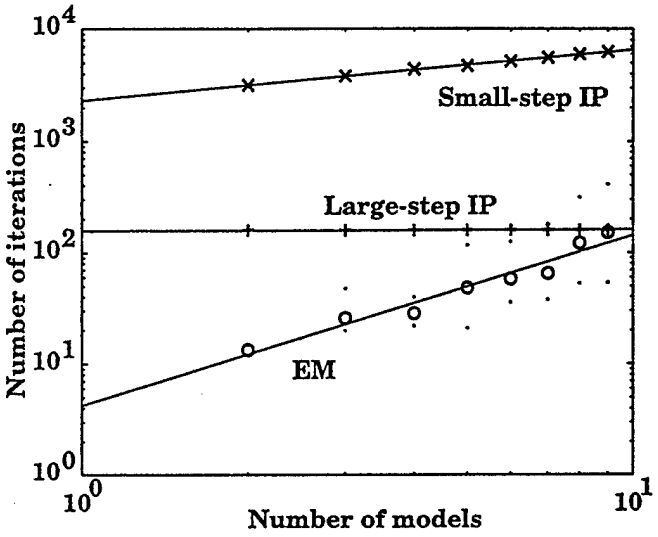


Figure 5: Algorithm iteration rates

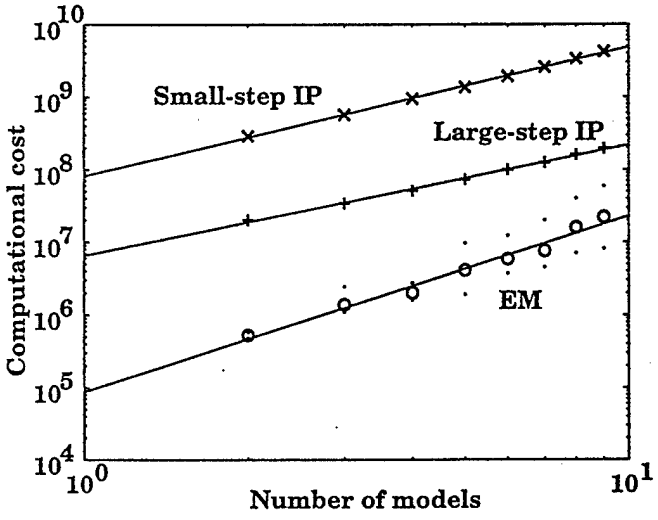


Figure 6: Algorithm computational costs

Table 1: Cost standard deviation results

Algorithm	N	% of costs > mean	% of costs within one standard deviation
EM	300	0.367	0.767
Large-step IP	300	0.417	0.717
EM	500	0.433	0.742
Large-step IP	500	0.467	0.700
EM	750	0.425	0.750
Large-step IP	750	0.400	0.700

exhibiting very similar mean costs for all three cases. The EM algorithm generally exhibits slightly better cost, but the IP algorithm outperforms the EM algorithm at some points.

Analysis of the costs' standard deviations reveals two observations. First, Table 1 tabulates the percentages for the two algorithms of the number of runs resulting in costs greater than the mean cost and the number of runs with costs within a standard deviation of the mean. The EM algorithm again performs slightly better in both accounts, but the results are comparable. Thus, the algorithms' statistical performances are comparable.

The second observation deals with the standard deviations, themselves. Figures 10-12 plot the standard deviations of the costs normalized by the mean cost for each number of mixtures for the three experiments. The results show that the IP algorithm consistently exhibits a smaller, normalized standard deviation than does EM. This implies that the IP algorithm requires a number of computations that vary less from run to run than does EM. In terms of algorithm operation, it appears that the IP algorithm is less susceptible to the effects of initial conditions in the number of required computations than is EM.

Final cost results examine the worst case costs incurred by the two algorithms. Figures 13-15 plot the maximum costs incurred by the two algorithms for the three test cases. In all three cases, the IP algorithm's worst case cost is generally better than the EM algorithm's worst case performance. Considering that EM has a cost with greater standard deviation than does the large-step method, this suggests that for these closely spaced problems the IP algorithm may be better in terms of cost incurred.

Error analysis, similar to the preceding cost analysis, follows here. Mean square parameter error results appear in Figures 16-18. The plots depict the mean square parameter error between the

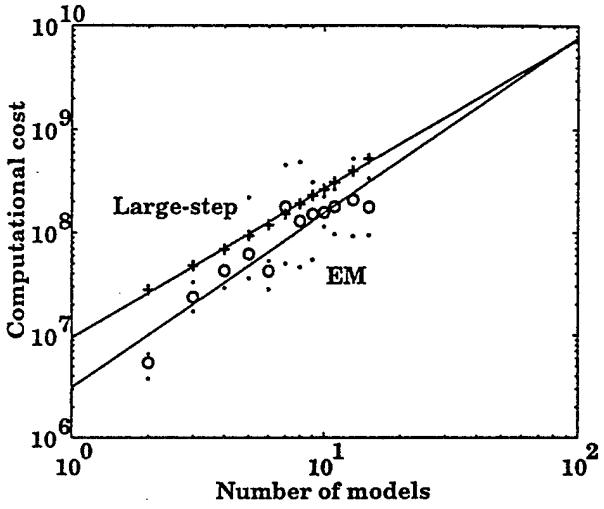


Figure 7: Computational cost ($N = 300$)

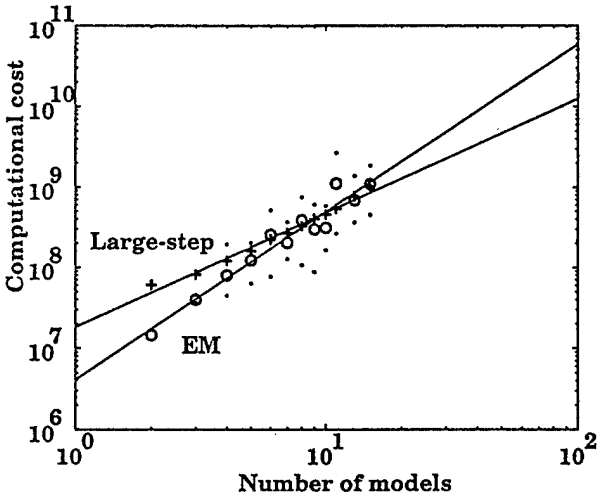


Figure 8: Computational cost ($N = 500$)

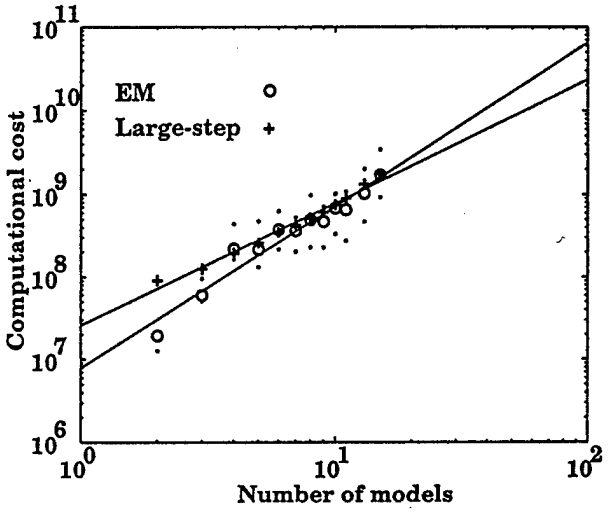


Figure 9: Computational cost ($N = 750$)

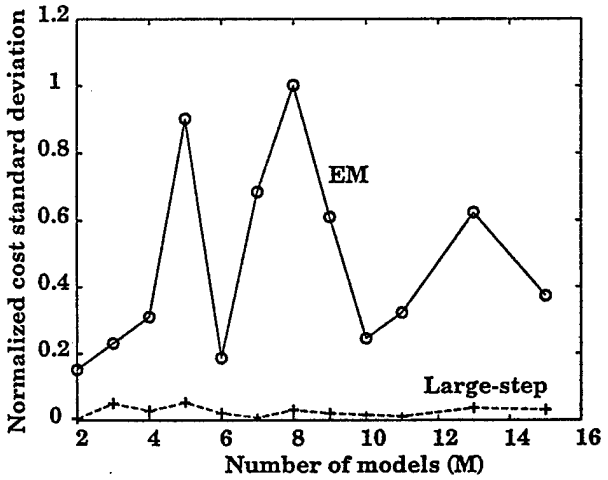


Figure 10: Normalized cost standard deviation ($N = 300$)

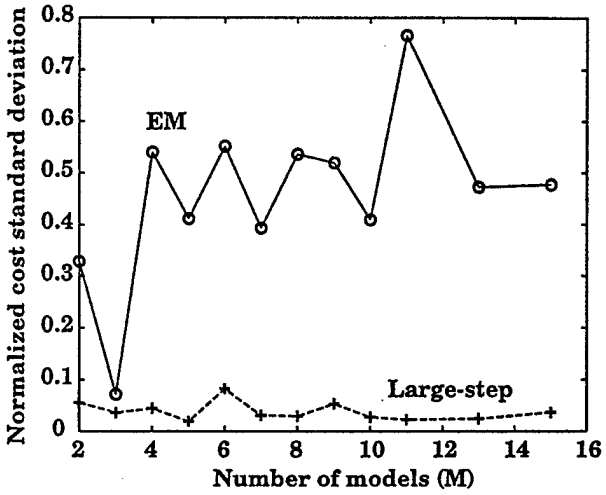


Figure 11: Normalized cost standard deviation ($N = 500$)

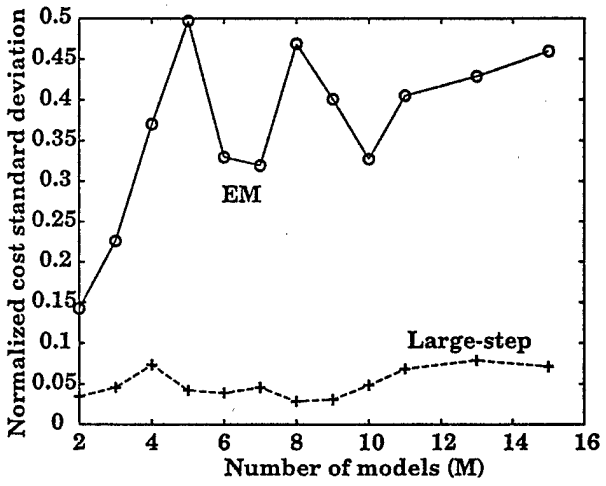


Figure 12: Normalized cost standard deviation ($N = 750$)

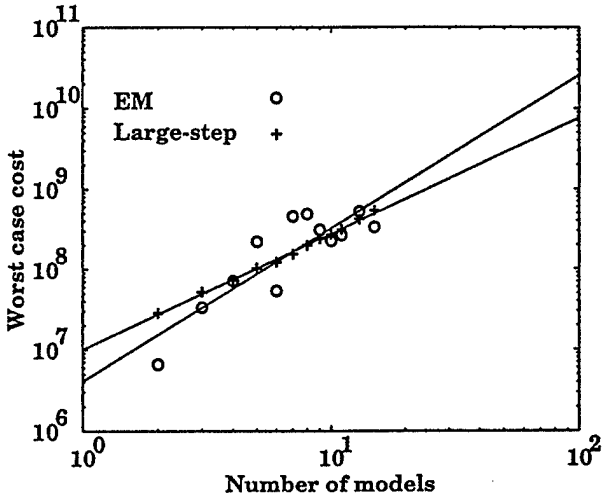


Figure 13: Worst case computational cost ($N = 300$)

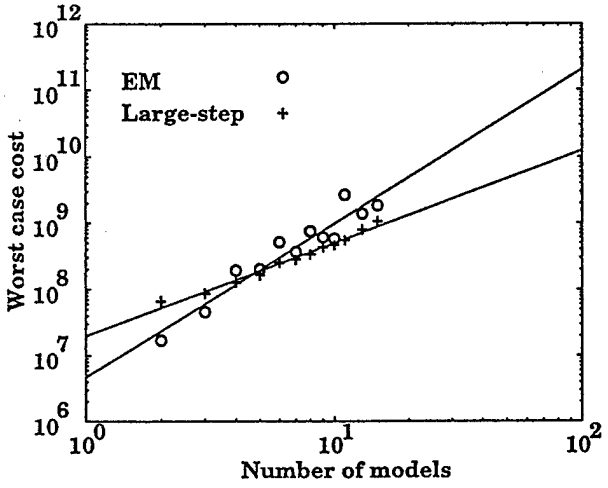


Figure 14: Worst case computational cost ($N = 500$)

Table 2: Error standard deviation results

Algorithm	N	% errors > mean	% costs within one standard deviation
EM	300	0.392	0.658
Large-step IP	300	0.517	0.667
EM	500	0.383	0.725
Large-step IP	500	0.483	0.667
EM	750	0.467	0.692
Large-step IP	750	0.467	0.683

computed estimates and the actual values for each number of models. The errors are averaged over the number of runs at each point. The plots demonstrate that the EM algorithm always results in worse error than does the IP algorithm.

Examination of the errors' standard deviation yields additional observations. Table 2 contains information similar to that contained in Table 1. It lists the percentages for the two algorithms of the number of runs resulting in errors greater than the mean error and the number of runs with errors that fall within a standard deviation of the mean error. This table suggests that the error distributions are comparable between both methods

Additional results in Figures 19-21 plot the standard deviations of the errors normalized by the mean error for each case. The plots demonstrate that the EM algorithm has a greater variance in the errors it incurs in producing its estimates for all three trials. The larger errors occur because the EM algorithm has more difficulty in separating the samples from run to run. The IP method, however, separates the samples with a much smaller variance in its estimates.

Final error results examine the worst case errors incurred by the two algorithms. Figures 22-24 plot the maximum errors incurred by the two algorithms. In all the cases, the IP algorithm's worst case error is significantly better than the EM algorithm's worst case error. This again supports the observation that the IP algorithm generalizes better than the EM algorithm on the test cases.

5.2 Model parameter identification

A final experiment examines the small-step HIP algorithm's use on a multiple model system identification problem. Here, a modular network is trained to identify multiple linear, set point models of a nonlinear system. The problems uses input/output data for the fuel flow process from a fossil fuel

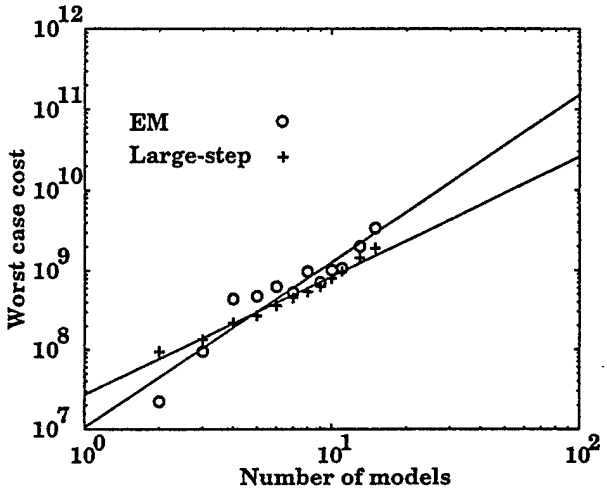


Figure 15: Worst case computational cost ($N = 750$)

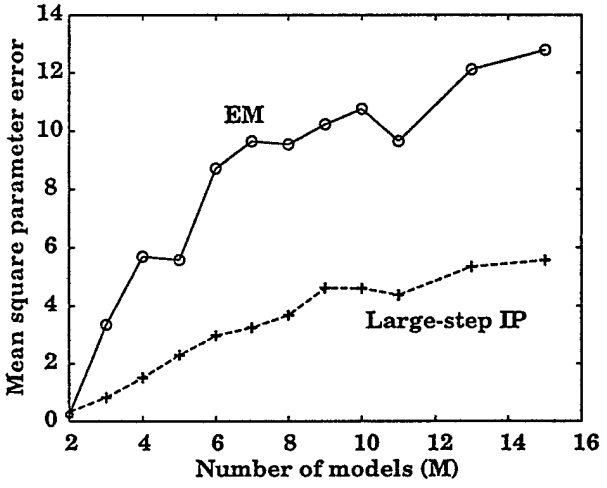


Figure 16: Parameter error($N = 300$)

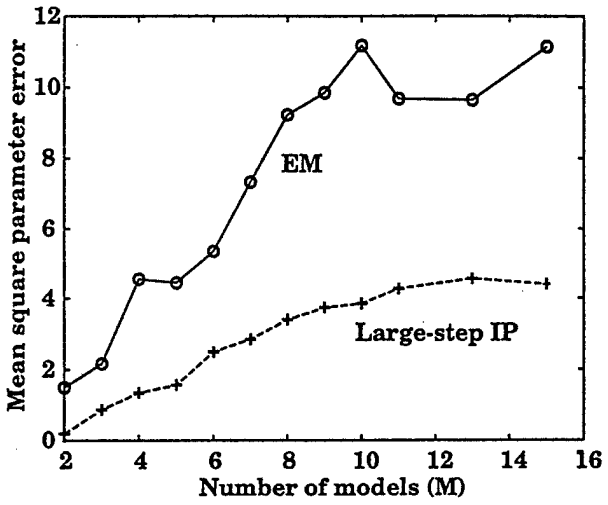


Figure 17: Parameter error ($N = 500$)

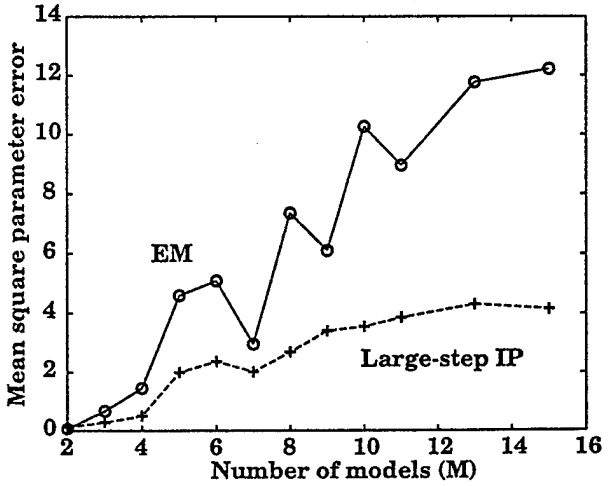


Figure 18: Parameter error ($N = 750$)

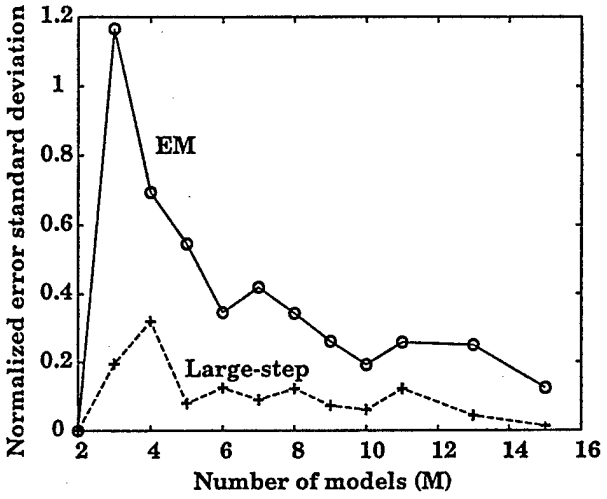


Figure 19: Normalized error standard deviation ($N = 300$)

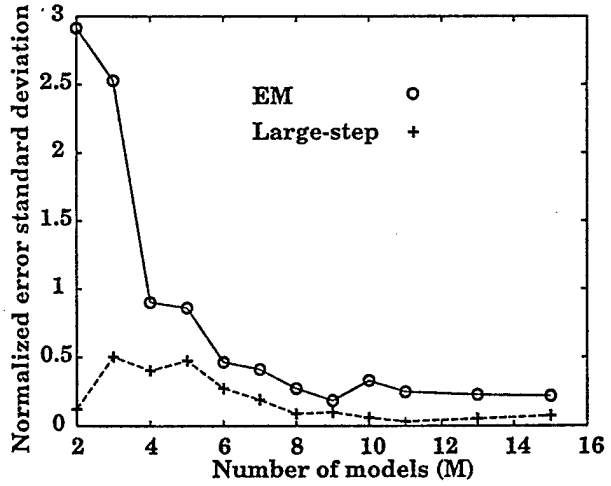


Figure 20: Normalized error standard deviation ($N = 500$)

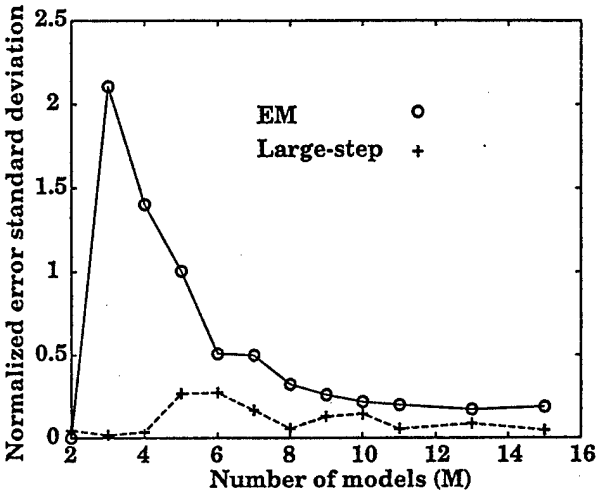


Figure 21: Normalized error standard deviation ($N = 750$)

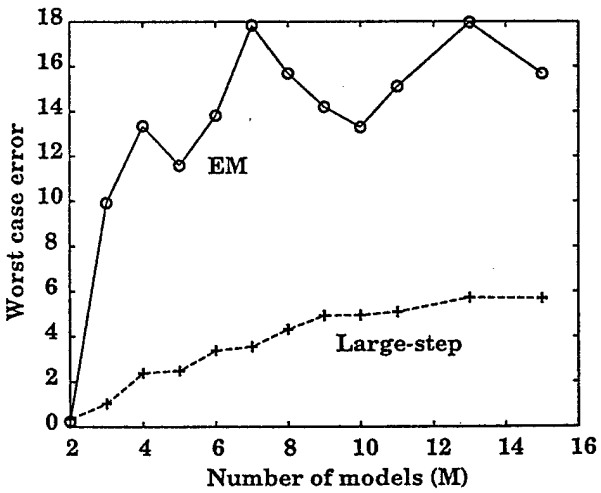


Figure 22: Worst case error ($N = 300$)

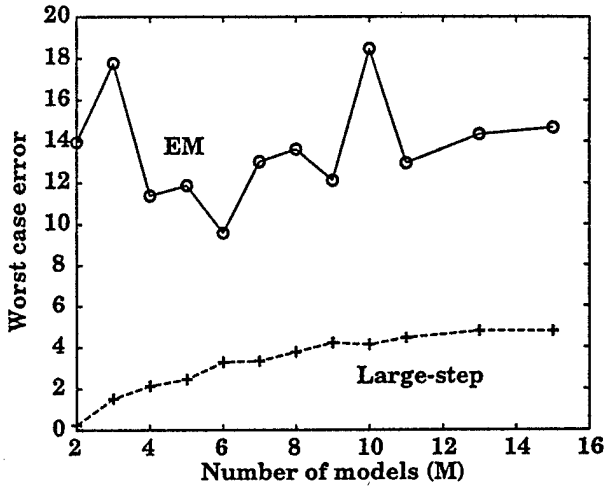


Figure 23: Worst case error ($N = 500$)

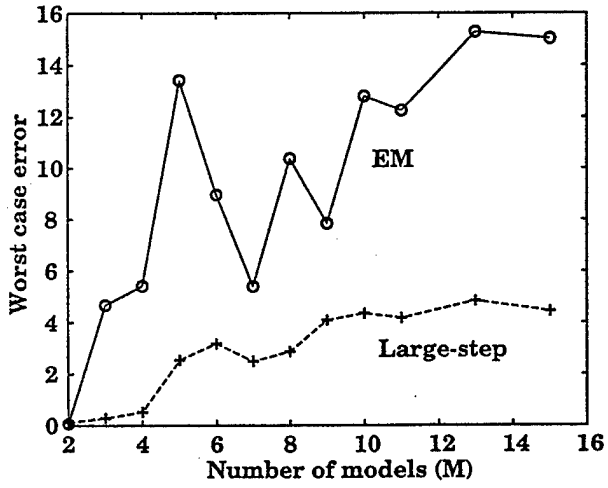


Figure 24: Worst case error ($N = 750$)

burning electric power generating plant (see Figure 25). Two sets of data consisting of input/output pairs of the fuel flow process are available to train the network: an $N = 300$ sample design set and an $N = 1438$ sample validation set. The experiment used the small-step HIP algorithm to train modular network and recorded the computational properties of the HIP algorithm for $M = 2$ to 9 model units. Also recorded was the network's performance in approximating the fuel flow process.

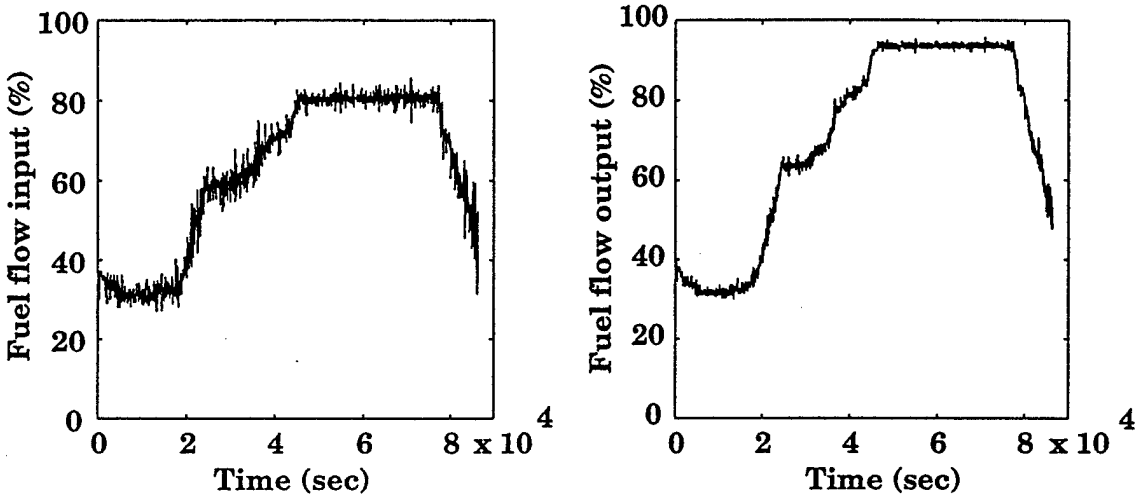


Figure 25: Fuel flow input/output data

The experiment's results follow here. Figure 26(a) shows that the empirical iteration rate is $O(n^{0.68})$ iterations and is slightly worse than the $O(\sqrt{n})$ predicted rate. Figure 26(b) shows that the $O(n^{1.9})$ flop empirical cost (open circles/solid line) exceeds the $O(n^{3.5})$ flop theoretical cost rate (dashed line). The difference between the theoretical and empirical rates in this and the previous experiments is attributed to the fact that sparse matrix techniques were used to implement the IP algorithms. Finally, figure 27 gives an indication of the resulting network's performance in approximating the fuel flow process. Figure 27(a) plots the mean square error of the approximation for hard and soft switching between models. The results show a decreasing and then slightly increasing trend with an optimum at $M = 4$ models. Figure 27 display the networks' performance for an $M = 4$ model approximation. The figure demonstrates that the resulting network approximates the fuel flow process well.

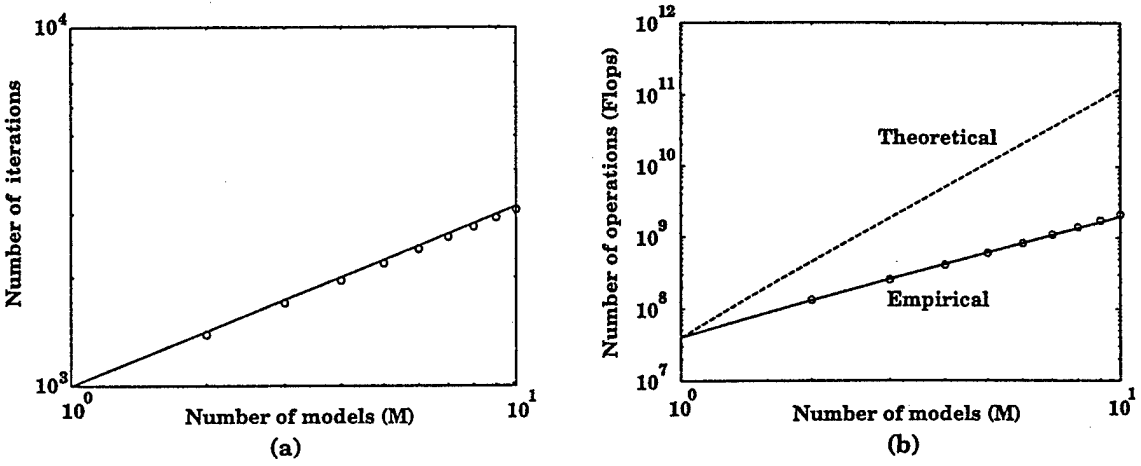


Figure 26: Fuel flow computational properties

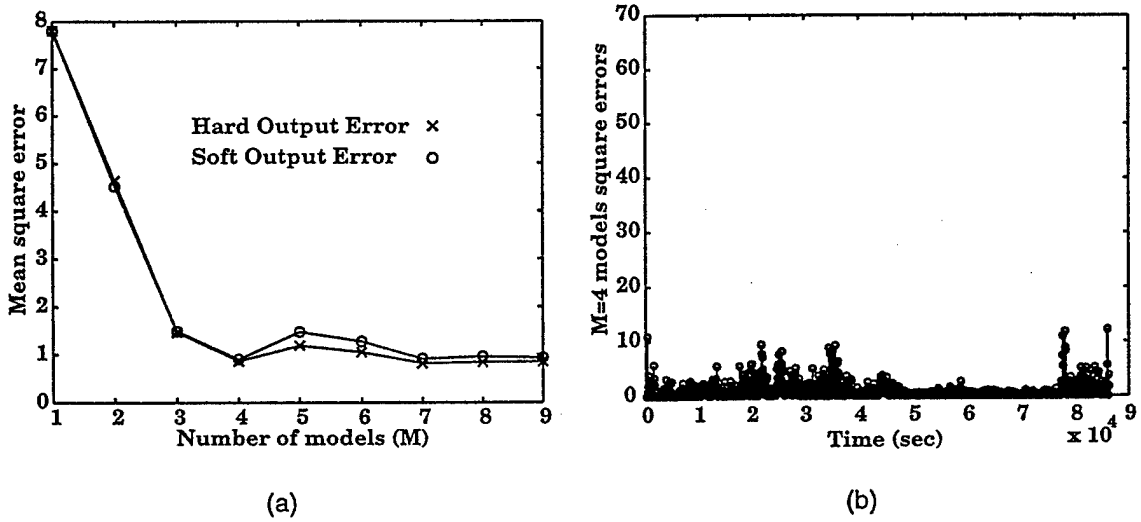


Figure 27: Fuel flow results

6 Discussion

This paper used the hybrid interior point (HIP) algorithm to train modular neural networks. The HIP procedure combines an interior-point linear programming algorithm with a Newton-Raphson iteration to produce an algorithm whose computational cost scales in a polynomial manner with problem size. In particular, this paper formally proves that the HIP algorithm converges to a fixed point that is a local optimum of the regularized training problem. This paper also shows how to configure the algorithm so that the number of iterations and total computational cost scale at the same rate as fast interior-point linear programming algorithms. The formal results obtained in this paper were validated through simulation experiments which compared large-step and small-step versions of the HIP algorithm against Expectation-Maxmimization (EM) procedures. These results showed that for the test suite of problems large-step HIP algorithms exhibited a total computational cost that was comparable to that of the EM procedures, but that the resulting solutions obtained by the HIP algorithm had lower errors than the EM procedure's solutions. The results also showed that the HIP algorithm's total computational costs were relatively insensitive to a problem's initial conditions; something which was definitely not the case with the EM methods. Further experiments demonstrated the use of the HIP algorithm in identifying multiple predictors for nonlinear dynamical plants.

A Appendix

A.1 Proximity proofs

Proof of Corollary 2: From [5], if $\delta(\bar{\mathbf{x}}^{(k)}, \alpha^{(k)}, \bar{\Theta}^{(k)}) \leq 0.5$ and $\alpha^{(k+1)} = \alpha^{(k)}(1 + \nu/\sqrt{n})$ with $\nu \in (0, 0.1]$, then $\delta(\bar{\mathbf{x}}^{(k)}, \alpha^{(k+1)}, \bar{\Theta}^{(k)}) \leq 0.7$. If $\delta(\bar{\mathbf{x}}^{(k)}, \alpha, \bar{\Theta}^{(k)}) < 1$, then one SSD step produces $\bar{\mathbf{x}}_1^{(k+1)}$ from $\bar{\mathbf{x}}^{(k)}$ where (Lemma 5.4 in [5]),

$$\delta(\bar{\mathbf{x}}_1^{(k+1)}, \alpha^{(k+1)}, \bar{\Theta}^{(k)}) \leq \delta(\bar{\mathbf{x}}^{(k)}, \alpha^{(k+1)}, \bar{\Theta}^{(k)})^2 \leq (0.7)^2.$$

Application of another SSD step produces $\bar{\mathbf{x}}_2^{(k+1)}$ from $\bar{\mathbf{x}}_1^{(k+1)}$ with

$$\delta(\bar{\mathbf{x}}_2^{(k+1)}, \alpha^{(k+1)}, \bar{\Theta}^{(k)}) \leq \delta(\bar{\mathbf{x}}_1^{(k+1)}, \alpha^{(k+1)}, \bar{\Theta}^{(k)})^2 \leq (0.7)^{2^2}.$$

Proceeding inductively, assuming that $\delta(\bar{\mathbf{x}}_J^{(k+1)}, \alpha^{(k+1)}, \bar{\Theta}^{(k)}) \leq (0.7)^{2^J}$, then

$$\delta(\bar{\mathbf{x}}_{J+1}^{(k+1)}, \alpha^{(k+1)}, \bar{\Theta}^{(k)}) \leq \delta(\bar{\mathbf{x}}_J^{(k+1)}, \alpha^{(k+1)}, \bar{\Theta}^{(k)})^2 \leq ((0.7)^{2^J})^2 = (0.7)^{2^{J+1}}$$

□

Proof of Theorem 4: The proof must demonstrate that using $\gamma^{(k)}$ in (27) maintains the nearness of $\bar{\mathbf{x}}^{(k+1)}$ to the central path after updating $\bar{\Theta}^{(k)}$. Let $\bar{\mathbf{h}}(\bar{\mathbf{x}}, \alpha, \bar{\Theta}) = P_{AX} X \left(\alpha \bar{\mathbf{c}}(\bar{\Theta}) - \bar{\mathbf{x}}^{-1}\right)$ and define $\bar{\mathbf{h}}_1$ and $\bar{\mathbf{h}}_2$ as $\bar{\mathbf{h}}_1 = \bar{\mathbf{h}}(\bar{\mathbf{x}}^{(k+1)}, \alpha^{(k+1)}, \bar{\Theta}^{(k)})$ and $\bar{\mathbf{h}}_2 = \bar{\mathbf{h}}(\bar{\mathbf{x}}^{(k+1)}, \alpha^{(k+1)}, \bar{\Theta}^{(k+1)})$. Using the triangle inequality, $\bar{\mathbf{h}}_1$ and $\bar{\mathbf{h}}_2$ are related as

$$\|\bar{\mathbf{h}}_2\| \leq \|\bar{\mathbf{h}}_2 - \bar{\mathbf{h}}_1\| + \|\bar{\mathbf{h}}_1\|.$$

Substituting $\bar{\mathbf{h}} = P_{AX} X \left(\alpha \bar{\mathbf{c}}(\bar{\Theta}) - \bar{\mathbf{x}}^{-1}\right)$ and δ_1 for $\|\bar{\mathbf{h}}_1\|$ produces

$$\|\bar{\mathbf{h}}_2\| \leq \|\alpha^{(k+1)} P_{AX^{(k+1)}} X^{(k+1)} (\bar{\mathbf{c}}^{(k+1)} - \bar{\mathbf{c}}^{(k)})\| + \delta_1.$$

Applying the quadratic update and using the non-negativity and convexity of $\bar{\mathbf{c}}(\bar{\Theta})$ ($\bar{\mathbf{c}}^{(k+1)} \leq (1 - \gamma^{(k)})\bar{\mathbf{c}}^{(k)} + \gamma^{(k)} \bar{\mathbf{c}}^{(k+1),*}$) results in

$$\|\bar{\mathbf{h}}_2\| \leq \|\alpha^{(k+1)} \gamma^{(k)} P_{AX^{(k+1)}} X^{(k+1)} (\bar{\mathbf{c}}^{(k+1),*} - \bar{\mathbf{c}}^{(k)})\| + \delta_1.$$

Using $\|P_{AX^{(k)}} X^{(k)}\| \leq \Delta^{(k)}$ from [15] and $\Delta^{(k)} \leq (n + 0.5\sqrt{n})/\alpha^{(k)} < 2n/\alpha^{(k)}$, then

$$\begin{aligned} \|\bar{\mathbf{h}}_2\| &\leq \gamma^{(k)} \alpha^{(k+1)} \|P_{AX^{(k+1)}} X^{(k+1)}\| \, \|\bar{\mathbf{c}}^{(k+1),*} - \bar{\mathbf{c}}^{(k)}\| + \delta_1. \\ &< \gamma^{(k)} 2n \|\bar{\mathbf{c}}^{(k+1),*} - \bar{\mathbf{c}}^{(k)}\| + \delta_1. \end{aligned}$$

Plugging in the value of γ from (27) yields

$$\|\bar{h}_2\| < \left[\frac{(\delta_2 - \delta_1)}{2n\|\bar{c}^{(k+1),*} - \bar{c}^{(k)}\|}\right] 2n\|\bar{c}^{(k+1),*} - \bar{c}^{(k)}\| + \delta_1 = \delta_2.$$

□

Proof of Theorem 5: By assumption, $\delta(\bar{x}^{(k)}, \alpha^{(k)}, \bar{\Theta}^{(k)}) \leq 0.5$. For the small-step algorithm, Corollary 2 states that two SSD steps produce $\bar{x}^{(k+1)}$ where $\delta(\bar{x}^{(k+1)}, \alpha^{(k+1)}, \bar{\Theta}^{(k)}) \leq (0.7)^{2^2} = 0.2401$. For the large-step algorithm, Theorem 3 states that $O(n\mu^2/(1+\mu^2))$ steps produce $\bar{x}^{(k+1)}$ such that $\delta(\bar{x}^{(k+1)}, \alpha^{(k+1)}, \bar{\Theta}^{(k)}) \leq 0.5$. Taking a constant two additional steps reduces $\delta(\bar{x}^{(k+1)}, \alpha^{(k+1)}, \bar{\Theta}^{(k)})$ below 0.2401. Thus, $O(n\mu^2/(1+\mu^2))$ SSD steps produce $\bar{x}^{(k+1)}$ such that $\delta(\bar{x}^{(k+1)}, \alpha^{(k+1)}, \bar{\Theta}^{(k)}) \leq 0.2401$. A valid $\gamma^{(k)}$ for the quadratic update can be computed using (27) where $\delta_1 = 0.2401$ and $\delta_2 = 0.5$, and the $\bar{\Theta}$-update results in $\bar{\Theta}^{(k+1)}$ such that

$$\delta(\bar{x}^{(k+1)}, \alpha^{(k+1)}, \bar{\Theta}^{(k+1)}) \leq 0.5.$$

□

A.2 Asymptotic convergence proofs

The convergence analysis begins with a set of definitions. Let $\Omega = (\bar{Q}, \bar{\Theta})$ be the set of solutions produced by the algorithm. Let $\Omega(\bar{Q}) \subset \Omega$ be the set of all $\bar{Q}$'s which are solutions to the LP problem. Let $\Omega(\bar{\Theta}) \subset \Omega$ be the set of all solutions to the quadratic problem. Define co(Ω) to be the convex hull of the set of points in Ω. Let γ be defined as

$$\gamma = \min\left(\frac{0.12995}{n\|\bar{c}(\bar{\Theta}_1) - \bar{c}(\bar{\Theta}_2)\|}, 1\right).$$

Lemma 13 *Let $\bar{\Theta}^{(k)}, \bar{\Theta}^{(k),*} \in co(\Omega(\bar{\Theta}))$ be the current and minimizing parameter vectors at iteration k. Let $\gamma^{(k)} \in (0,1]$ be defined as in (27) and let the parameter vectors be updated as*

$$\bar{\Theta}^{(k+1)} = (1 - \gamma^{(k)})\bar{\Theta}^{(k)} + \gamma^{(k)}\bar{\Theta}^{(k+1),*}. \tag{36}$$

If

$$\|\bar{\Theta}^{(k+2),*} - \bar{\Theta}^{(k+1),*}\| < \gamma^{(k)}\|\bar{\Theta}^{(k+1),*} - \bar{\Theta}^{(k)}\|, \tag{37}$$

then

$$\|\bar{\Theta}^{(k+2),*} - \bar{\Theta}^{(k+1)}\| < \|\bar{\Theta}^{(k+1),*} - \bar{\Theta}^{(k)}\|. \tag{38}$$

Proof: Begin by adding and subtracting $\bar{\Theta}^{(k+1),*}$ to $\|\bar{\Theta}^{(k+2),*} - \bar{\Theta}^{(k+1)}\|$ and then using the triangle inequality to break apart the norm

$$\|\bar{\Theta}^{(k+2),*} - \bar{\Theta}^{(k+1)}\| \leq \|\bar{\Theta}^{(k+2),*} - \bar{\Theta}^{(k+1),*}\| + \|\bar{\Theta}^{(k+1),*} - \bar{\Theta}^{(k+1)}\|.$$

Using the expression for $\bar{\Theta}^{(k+1)}$ in (36) and then substituting (37) into the result produces the desired result

$$\begin{aligned} \|\bar{\Theta}^{(k+2),*} - \bar{\Theta}^{(k+1)}\| &\leq \|\bar{\Theta}^{(k+2),*} - \bar{\Theta}^{(k+1),*}\| + (1-\gamma^{(k)})\|\bar{\Theta}^{(k+1),*} - \bar{\Theta}^{(k)}\| \\ &< \gamma^{(k)}\|\bar{\Theta}^{(k+1),*} - \bar{\Theta}^{(k)}\| + (1-\gamma^{(k)})\|\bar{\Theta}^{(k+1),*} - \bar{\Theta}^{(k)}\| \\ &< \|\bar{\Theta}^{(k+1),*} - \bar{\Theta}^{(k)}\| \end{aligned}$$

□

Proof of Theorem 6: Starting with $(\bar{\mathbf{x}}^{(k)}, \alpha^{(k)})$, updating $\alpha^{(k)}$, and performing two SSD steps as

$$\begin{aligned} \bar{\mathbf{x}}_1 &= \bar{\mathbf{x}}^{(k)} - X^{(k)} P_{AX^{(k)}} X^{(k)} (\alpha^{(k+1)} \bar{\mathbf{c}}^{(k)} - (\bar{\mathbf{x}}^{(k)})^{-1}) \\ \bar{\mathbf{x}}^{(k+1)} &= \bar{\mathbf{x}}_1 - X_1 P_{AX_1} X_1 (\alpha^{(k+1)} \bar{\mathbf{c}}^{(k)} - \bar{\mathbf{x}}_1^{-1}) \end{aligned}$$

where $X^{(k)} = \text{diag}(\bar{\mathbf{x}}^{(k)}) \in \mathbb{R}^{n \times n}$ and $X_1 = \text{diag}(\bar{\mathbf{x}}_1) \in \mathbb{R}^{n \times n}$ results in $\bar{\mathbf{x}}^{(k+1)}$. Let

$$\begin{aligned} \|\Delta Q\| &= \|\bar{\mathbf{x}}^{(k+1)} - \bar{\mathbf{x}}^{(k)}\| \\ &\leq \|\bar{\mathbf{x}}^{(k+1)} - \bar{\mathbf{x}}_1\| + \|\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}^{(k)}\|. \end{aligned}$$

Examine the distance moved in the first update, $\|\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}^{(k)}\|$. Use the facts that $\alpha^{(k+1)} = \alpha(1+\nu/\sqrt{n})$ and that P_{AX} is an orthogonal projection ($P_{AX} = P_{AX}^2$).

$$\begin{aligned} \|\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}^{(k)}\| &= \left\| X^{(k)} P_{AX^{(k)}} X^{(k)} (\alpha^{(k+1)} \bar{\mathbf{c}}^{(k)} - (\bar{\mathbf{x}}^{(k)})^{-1}) \right\| \\ &= \|X^{(k)} P_{AX^{(k)}}^2 X^{(k)} \left[(\alpha^{(k)} \bar{\mathbf{c}}^{(k)} - (\bar{\mathbf{x}}^{(k)})^{-1}) + \alpha^{(k)} \nu \bar{\mathbf{c}}^{(k)} / \sqrt{n} \right] \| \\ &\leq \left\| X^{(k)} P_{AX^{(k)}} \right\| \left\| P_{AX^{(k)}} X^{(k)} \left[(\alpha^{(k)} \bar{\mathbf{c}}^{(k)} - (\bar{\mathbf{x}}^{(k)})^{-1}) + \alpha^{(k)} \nu \bar{\mathbf{c}}^{(k)} / \sqrt{n} \right] \right\| \end{aligned}$$

But $XP_{AX} = P_{AX}X$ and $\|P_{AX^{(k)}} X^{(k)}\| \leq \Delta^{(k)}$ [15] so

$$\|\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}^{(k)}\| \leq \left\| P_{AX^{(k)}} X^{(k)} \left[(\alpha^{(k)} \bar{\mathbf{c}}^{(k)} - (\bar{\mathbf{x}}^{(k)})^{-1}) + \alpha^{(k)} \nu \bar{\mathbf{c}}^{(k)} / \sqrt{n} \right] \right\| \Delta^{(k)}$$

Using (23), $\alpha^{(k)} \frac{\nu}{\sqrt{n}} \|P_{AX^{(k)}} X^{(k)} \bar{\mathbf{c}}^{(k)}\| \leq \nu(\delta + 1)$ [5], and letting $\delta = \delta(\bar{\mathbf{x}}^{(k)}, \alpha^{(k)}, \bar{\Theta}^{(k)})$, the result becomes

$$\begin{aligned} \|\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}^{(k)}\| &\leq [\delta + \nu(\delta + 1)] \Delta^{(k)} \\ &\leq (\delta(1+\nu) + \nu) \Delta^{(k)} \end{aligned}$$

Now examine the distance moved in the second update, $\|\bar{\mathbf{x}}^{(k+1)} - \bar{\mathbf{x}}_1\|$

$$\begin{aligned}\|\bar{\mathbf{x}}^{(k+1)} - \bar{\mathbf{x}}_1\| &= \left\|X_1 P_{AX_1} X_1(\alpha^{(k+1)}\bar{\mathbf{c}}^{(k)} - \bar{\mathbf{x}}_1^{-1})\right\| \\ &= \|X_1 P_{AX_1}^2 X_1(\alpha^{(k+1)}\bar{\mathbf{c}}^{(k)} - \bar{\mathbf{x}}_1^{-1})\| \\ &\leq \|P_{AX_1} X_1(\alpha^{(k+1)}\bar{\mathbf{c}}^{(k)} - \bar{\mathbf{x}}_1^{-1})\|\Delta^{(k)}\end{aligned}$$

However, $\|P_{AX_1} X_1(\alpha^{(k+1)}\bar{\mathbf{c}}^{(k)} - \bar{\mathbf{x}}_1^{-1})\| = \delta(\bar{\mathbf{x}}_1, \alpha^{(k+1)}, \bar{\Theta}^{(k)})$. Using Lemma 5.4 in [5], it can be shown that $\delta(\bar{\mathbf{w}}^{(k+1)}, \alpha^{(k+1)}, \bar{\Theta}^{(k)}) \leq \delta(\bar{\mathbf{w}}^{(k)}, \alpha^{(k+1)}, \bar{\Theta}^{(k)})^2$. Since $\delta(\bar{\mathbf{x}}^{(k)}, \alpha^{(k+1)}, \bar{\Theta}^{(k)}) = \|P_{AX^{(k)}} X^{(k)}(\alpha^{(k+1)}\bar{\mathbf{c}}^{(k)} - (\bar{\mathbf{x}}^{(k)})^{-1})\| = (\delta(1+\nu)+\nu)$ from above, then $\delta(\bar{\mathbf{x}}_1, \alpha^{(k+1)}, \bar{\Theta}^{(k)}) \leq (\delta(1+\nu)+\nu)^2$. Thus,

$$\begin{aligned}\|\bar{\mathbf{x}}^{(k+1)} - \bar{\mathbf{x}}_1\| &\leq \delta(\bar{\mathbf{x}}_1, \alpha^{(k+1)}, \bar{\Theta}^{(k)})\Delta^{(k)} \\ &\leq (\delta(1+\nu)+\nu)^2\Delta^{(k)}\end{aligned}$$

But $(\delta(1+\nu)+\nu) < 1$ and $(\delta(1+\nu)+\nu)^2 < (\delta(1+\nu)+\nu)$, therefore

$$\begin{aligned}\|\Delta Q\| &\leq \|\bar{\mathbf{x}}^{(k+1)} - \bar{\mathbf{x}}_1\| + \|\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}^{(k)}\| \\ &\leq (\delta(1+\nu)+\nu)^2\Delta^{(k)} + (\delta(1+\nu)+\nu)\Delta^{(k)} \\ &< 2(\delta(1+\nu)+\nu)\Delta^{(k)}\end{aligned}$$

□

Proof of Theorem 7: Let $\bar{\Theta} = \langle \bar{\theta}_1^T, \bar{\theta}a_2^T, \ldots, \bar{\theta}a_M^T\rangle^T$ and let $\bar{\theta}a_m = \langle \bar{\phi}_m^T, \bar{\omega}_m^T\rangle^T$, then

$$\begin{aligned}\|\bar{\Theta}^{(k+1),*} - \bar{\Theta}^{(k),*}\| &\leq \textstyle\sum_{m=1}^{M} \|\bar{\theta}_m^{(k+1),*} - \bar{\theta}_m^{(k),*}\| \\ &\leq \textstyle\sum_{m=1}^{M} \|\bar{\phi}_m^{(k+1),*} - \bar{\phi}_m^{(k),*}\| + \|\bar{\omega}_m^{(k+1),*} + \bar{\omega}_m^{(k),*}\|.\end{aligned}$$

Let

$$\begin{aligned}\bar{\mathbf{w}} &= \textstyle\sum_{j=1}^{N} \frac{1}{N} Q(m|j) y_j \bar{\mathbf{z}}_j \\ \Delta\bar{\mathbf{w}} &= \textstyle\sum_{j=1}^{N} \frac{1}{N} \Delta Q(m|j) y_j \bar{\mathbf{z}}_j\end{aligned}$$

where $\bar{\mathbf{w}}, \Delta\bar{\mathbf{w}} \in \mathbb{R}^R$. Using the definitions for $Q^{(k)}(m|j)$, $Q^{(k+1)}(m|j)$, $\bar{\phi}_m^*$, and $\bar{\omega}_m^*$, one finds that

$$\bar{\phi}_m^{(k+1),*} - \bar{\phi}_m^{(k),*} = (A_m + E_m)^{-1}(\bar{\mathbf{w}} + \Delta\bar{\mathbf{w}}) - A_m^{-1}\bar{\mathbf{w}}_m$$

$$\bar{\omega}_m^{(k+1),*} - \bar{\omega}_m^{(k),*} = \frac{\sum_{j=1}^{N} \frac{1}{N}(Q(m|j) + \Delta Q(m|j))\bar{\mathbf{z}}_j}{\sum_{j=1}^{N} \frac{1}{N}(Q(m|j) + \Delta Q(m|j))} - \frac{\sum_{j=1}^{N} \frac{1}{N} Q(m|j)\bar{\mathbf{z}}_j}{\sum_{j=1}^{N} \frac{1}{N} Q(m|j)}$$

Grouping like terms, taking the norms of the differences, and using the triangle inequality

$$\|\bar{\phi}_m^{(k+1),*} - \bar{\phi}_m^{(k),*}\| \leq \|(A_m + E_m)^{-1} - A_m^{-1}\|\,\|\bar{\mathbf{w}}_m\| + \|(A_m + E_m)^{-1}\|\,\|\Delta\bar{\mathbf{w}}_m\|$$

$$\begin{aligned}\|\bar{\omega}_m^{(k+1),*} - \bar{\omega}_m^{(k),*}\| \leq &\left\|\left[\frac{\sum_{j=1}^{N} \frac{1}{N} Q(m|j)\bar{\mathbf{z}}_j}{\sum_{j=1}^{N} \frac{1}{N}(Q(m|j)+\Delta Q(m|j))} - \frac{\sum_{j=1}^{N} \frac{1}{N} Q(m|j)\bar{\mathbf{z}}_j}{\sum_{j=1}^{N} \frac{1}{N} Q(m|j)}\right]\right\| \\ &+ \left\|\frac{\sum_{j=1}^{N} \frac{1}{N} \Delta Q(m|j)\bar{\mathbf{z}}_j}{\sum_{j=1}^{N} \frac{1}{N}(Q(m|j)+\Delta Q(m|j))}\right\|\end{aligned}$$

The triangle inequality and the properties of norms can be used to show that $\|\Delta\bar{\mathbf{w}}_m\| \leq \zeta Y_{max}\|\Delta Q\|$, $\|\bar{\mathbf{w}}\| \leq \zeta Y_{max}$, and $\|\sum_{j=1}^{N} \frac{1}{N} Q(m|j)\bar{\mathbf{z}}_j\| \leq \zeta$. Thus

$$\|\bar{\phi}_m^{(k+1),*} - \bar{\phi}_m^{(k),*}\| \leq \zeta Y_{max}\|(A_m + E_m)^{-1} - A_m^{-1}\| + \zeta Y_{max}\|\Delta Q\|\|(A_m + E_m)^{-1}\|$$

$$\|\bar{\omega}_m^{(k+1),*} - \bar{\omega}_m^{(k),*}\| \leq \zeta \left\| \left[\frac{1}{\sum_{j=1}^{N} \frac{1}{N}(Q(m|j)+\Delta Q(m|j))} - \frac{1}{\sum_{j=1}^{N} \frac{1}{N} Q(m|j)} \right] \right\| + \frac{\zeta\|\Delta Q\|}{\|\sum_{j=1}^{N} \frac{1}{N}(Q(m|j)+\Delta Q(m|j))\|}$$

$[A_m + E_m] = \sum_{j=1}^{N} \frac{1}{N} Q^{(k+1)}(m|j)\bar{\mathbf{z}}_j\bar{\mathbf{z}}_j^T$ is full rank by assumption and $\|(A_m + E_m)^{-1}\| \leq \mu_{max}$ and $(\sum_{j=1}^{N} \frac{1}{N} Q^{(k+1)}(m|j))^{-1} \leq \beta_{max}$, so

$$\|\bar{\phi}_m^{(k+1),*} - \bar{\phi}_m^{(k),*}\| \leq \zeta Y_{max}\|(A_m + E_m)^{-1} - A_m^{-1}\| + \mu_{max}\zeta Y_{max}\|\Delta Q\|$$

$$\|\bar{\omega}_m^{(k+1),*} - \bar{\omega}_m^{(k),*}\| \leq \zeta \left\| \frac{1}{\sum_{j=1}^{N} \frac{1}{N}(Q(m|j)+\Delta Q(m|j))} - \frac{1}{\sum_{j=1}^{N} \frac{1}{N} Q(m|j)} \right\| + \beta_{max}\zeta\|\Delta Q\|$$

Using Theorem 2.3.4 from [4],

$$\|(A_m + E_m)^{-1} - A_m^{-1}\| \leq \|E_m\|\mu_{max}^2/(1 - r_1)$$

and

$$\left\| \frac{1}{\sum_{j=1}^{N} \frac{1}{N}(Q(m|j) + \Delta Q(m|j))} - \frac{1}{\sum_{j=1}^{N} \frac{1}{N} Q(m|j)} \right\| \leq \beta_{max}^2\|\Delta Q\|/(1 - r_2)$$

where $\|E_m\| \leq \zeta^2\|\Delta Q\|$ the bound becomes

$$\|\bar{\phi}_m^{(k+1),*} - \bar{\phi}_m^{(k),*}\| \leq \mu_{max}^2\zeta^3 Y_{max}\|\Delta Q\|/(1 - r_1) + \mu_{max}\zeta Y_{max}\|\Delta Q\|$$

$$\|\bar{\omega}_m^{(k+1),*} - \bar{\omega}_m^{(k),*}\| \leq \beta_{max}^2\zeta\|\Delta Q\|/(1 - r_2) + \beta_{max}\zeta\|\Delta Q\|$$

Substituting these results into the expression for $\|\bar{\Theta}^{(k+1),*} - \bar{\Theta}^{(k),*}\|$ produces

$$\|\bar{\Theta}^{(k+1),*} - \bar{\Theta}^{(k),*}\| \leq MK\|\Delta Q\|$$

□

Proof of Theorem 8: Let $\mathcal{A}$ be the set

$$\mathcal{A} = \left\{ \text{co}(\Omega(\bar{\Theta})) \times \text{co}(\Omega(\bar{\Theta})) : \|\bar{\mathbf{c}}(\bar{\Theta}_1) - \bar{\mathbf{c}}(\bar{\Theta}_2)\| \geq \frac{0.12995}{n} \right\}.$$

$\mathcal{A}$ corresponds to all pairs of $\bar{\Theta}$'s in $\text{co}(\Omega(\bar{\Theta})) \times \text{co}(\Omega(\bar{\Theta}))$ that result in γ being less than or exactly equal to unity. Further define $\epsilon_{\bar{\Theta}}$ as

$$\epsilon_{\bar{\Theta}} = \min_{(\bar{\Theta}_1, \bar{\Theta}_2) \in \mathcal{A}} \gamma\|\bar{\Theta}_1 - \bar{\Theta}_2\|. \tag{39}$$

$\epsilon_{\bar{\Theta}}$ corresponds to the minimum value of $\gamma\|\bar{\Theta}_1 - \bar{\Theta}_2\|$ for all pairs of $\bar{\Theta}$'s which result in γ being less than or exactly equal to unity. If the set, $\mathcal{A}$, is empty, then $\epsilon_{\bar{\Theta}}$ is defined to be ∞ as all values of $(\bar{\Theta}_1, \bar{\Theta}_2) \in \text{co}(\bar{\Theta}) \times \text{co}(\bar{\Theta})$ for the problem are such that $\gamma^{(k)} = 1$. $\epsilon_{\bar{\Theta}}$ is bounded away from zero as $\text{co}(\bar{\Theta})$ is a bounded, convex set which means that there is a maximum $\|\bar{\Theta}_1 - \bar{\Theta}_2\|$ and $\|\bar{\mathbf{c}}(\bar{\Theta}_1) - \bar{\mathbf{c}}(\bar{\Theta}_2)\|$ for the set. Let K_1 be the smallest k for which

$$\|\bar{\Theta}^{(k+2),*} - \bar{\Theta}^{(k+1),*}\| < 2(\delta(1+\nu)+\nu)M\mathcal{K}\Delta^{(k)} < \epsilon_{\bar{\Theta}}. \tag{40}$$

K_1 exists as $\epsilon_{\bar{\Theta}} > 0$ and $\Delta^{(k)} \to 0$ with increasing k. The inequality with $k = K_1$ signifies the iteration, K_1, at which $\|\bar{\Theta}^{(k+2),*} - \bar{\Theta}^{(k+1),*}\|$ is smaller than the minimum possible term, $\gamma\|\bar{\Theta}_1 - \bar{\Theta}_2\|$. It is at K_1 that the sequence, $\|\bar{\Theta}^{(k+1),*} - \bar{\Theta}^{(k)}\|$ becomes strictly decreasing.

Examine $\gamma^{(k)}$ for the following two cases where $k \geq K_1$.

$\gamma^{(k)} < 1$: Equation (40) implies that

$$2(\delta(1+\nu)+\nu)M\mathcal{K}\Delta^{(k)} < \gamma^{(k)}\|\bar{\Theta}^{(k+1),*} - \bar{\Theta}^{(k)}\|$$

as $2(\delta(1+\nu)+\nu)M\mathcal{K}\Delta^{(k)} < \epsilon_{\bar{\Theta}}$ and $\epsilon_{\bar{\Theta}} \leq \gamma^{(k)}\|\bar{\Theta}^{(k+1),*} - \bar{\Theta}^{(k)}\|$ by its definition. Since $\|\bar{\Theta}^{(k+2),*} - \bar{\Theta}^{(k+1),*}\| < \gamma^{(k)}\|\bar{\Theta}^{(k+1),*} - \bar{\Theta}^{(k)}\|$ and $\gamma^{(k)} < 1$, then by Lemma 13, $\|\bar{\Theta}^{(k+1),*} - \bar{\Theta}^{(k)}\|$ is a strictly decreasing sequence. Let

$$\bar{\mathcal{A}} = \left\{ \text{co}(\Omega(\bar{\Theta})) \times \text{co}(\Omega(\bar{\Theta})) : \|\bar{\mathbf{c}}(\bar{\Theta}_1) - \bar{\mathbf{c}}(\bar{\Theta}_2)\| = \frac{0.12995}{n} \right\}.$$

be the set of all pairs of $\bar{\Theta}$'s in $\text{co}(\Omega(\bar{\Theta})) \times \text{co}(\Omega(\bar{\Theta}))$ that result in $\gamma^{(k)}$ being exactly equal to unity. Define the term, a, as

$$a = \min_{(\bar{\Theta}_1, \bar{\Theta}_2) \in \bar{\mathcal{A}}} \|\bar{\Theta}_1 - \bar{\Theta}_2\|$$

a is the minimum separation between parameter vectors in $\text{co}(\Omega(\bar{\Theta}))$ that results in $\gamma^{(k)}$ being exactly equal to one. a is greater than or equal to $\epsilon_{\bar{\Theta}}$. Since $\|\bar{\Theta}^{(k+1),*} - \bar{\Theta}^{(k)}\|$ is strictly decreasing to zero as it is bounded above by $\Delta^{(k)}$ which decreases to zero, it eventually decreases past a at which point $\gamma^{(k)} = 1$. Let the iteration at which $\|\bar{\Theta}^{(k+1),*} - \bar{\Theta}^{(k)}\| < a$ be $k = K_2$.

$\gamma^{(k)} = 1$: The first case showed that $\gamma^{(k)} \to 1$ for $k \geq K_1$. This second case demonstrates that $\gamma^{(k)} = 1$ for all $k \geq K_2 \geq K_1$. Let $k \geq K_2$ and $\gamma^{(k)} = 1$. Assume that $\gamma^{(k+1)} < 1$. Now, $\gamma^{(k)} = 1$ implies that $\bar{\Theta}^{(k+1)} = (1-\gamma^{(k)})\bar{\Theta}^{(k)} + \gamma^{(k)}\bar{\Theta}^{(k+1),*} = \bar{\Theta}^{(k+1),*}$. $\gamma^{(k+1)} < 1$ implies that

$\|\bar{\Theta}^{(k+2),*} - \bar{\Theta}^{(k+1)}\| = \|\bar{\Theta}^{(k+2),*} - \bar{\Theta}^{(k+1),*}\| > a$. But by assumption, $\|\bar{\Theta}^{(k+2),*} - \bar{\Theta}^{(k+1),*}\| < \epsilon_{\bar{\Theta}}$ and $\epsilon_{\bar{\Theta}} < a$ by definition. This is a contradiction, so the assumption is incorrect and $\gamma^{(k+1)} = 1$.

Thus, for $k \geq K = K_2 \geq K_1 \geq 0$, $\gamma^{(k)} = 1$. □

Proof of Theorem 9: Let the k^{th} solution be $\bar{\mathbf{w}}^{(k)} = \langle(\bar{\mathbf{x}}^{(k)})^T, (\bar{\Theta}^{(k)})^T\rangle^T$ and define the term, $\bar{\mathbf{w}}^* = \langle(\bar{\mathbf{x}}^*)^T, (\bar{\Theta}^*)^T\rangle^T$. Begin by examining $\|\bar{\mathbf{w}}^{(k+1)} - \bar{\mathbf{w}}^{(k)}\|$ for $k \geq K$.

$$\begin{aligned}
\|\bar{\mathbf{w}}^{(k+1)} - \bar{\mathbf{w}}^{(k)}\| &\leq \|\bar{\mathbf{x}}^{(k+1)} - \bar{\mathbf{x}}^{(k)}\| + \|\bar{\Theta}^{(k+1),*} - \bar{\Theta}^{(k)}\| \\
&\leq \|\bar{\mathbf{x}}^{(k+1)} - \bar{\mathbf{x}}^{(k)}\| + \|\bar{\Theta}^{(k+1),*} - \bar{\Theta}^{(k),*}\| \\
&< 2(\delta(1+\nu)+\nu)\Delta^{(k)} + M\mathcal{K}2(\delta(1+\nu)+\nu)\Delta^{(k)} \\
&< 2(\delta(1+\nu)+\nu)(1+M\mathcal{K})\Delta^{(k)}
\end{aligned}$$

Examine the following sum for all $j \geq k \geq K$

$$\sum_{j=k}^{\infty} \|\bar{\mathbf{w}}^{(j+1)} - \bar{\mathbf{w}}^{(j)}\| < \sum_{j=k}^{\infty} 2(\delta(1+\nu)+\nu)(1+M\mathcal{K})\Delta^{(j)} \tag{41}$$

The duality gap, $\Delta^{(k)}$, can be expressed as $\Delta^{(k)} = \beta^k \Delta^{(0)}$ where $\beta = 1/(1+\nu/\sqrt{n})$ [5]. Using this fact, (41) can be simplified as

$$\begin{aligned}
\textstyle\sum_{j=k}^{\infty} \|\bar{\mathbf{w}}^{(j+1)} - \bar{\mathbf{w}}^{(j)}\| &< \textstyle\sum_{j=k}^{\infty} 2(\delta(1+\nu)+\nu)(1+M\mathcal{K})\Delta^{(j)} \\
&< \textstyle\sum_{j=k}^{\infty} 2(\delta(1+\nu)+\nu)(1+M\mathcal{K})\beta^j\Delta^{(0)} \\
&< 2(\delta(1+\nu)+\nu)(1+M\mathcal{K})\frac{\Delta^{(k)}}{1-\beta}
\end{aligned}$$

This result corresponds to a constant, $2(\delta(1+\nu)+\nu)(1+M\mathcal{K})/(1-\beta)$, multiplied by the strictly decreasing duality gap, $\Delta^{(k)}$. Let k be chosen so that $\sum_{j=k}^{\infty} \|\bar{\mathbf{w}}^{(j+1)} - \bar{\mathbf{w}}^{(j)}\| < \epsilon$ for $\epsilon > 0$ (i.e. let the duality gap be small enough so that the condition holds). If k is so chosen, then $\forall i, j \geq k$, $\|\bar{\mathbf{w}}^{(i)} - \bar{\mathbf{w}}^{(j)}\| < \epsilon$ for arbitrary ϵ. Thus, $\{\bar{\mathbf{w}}^{(k)}\}$ is a Cauchy sequence; and it converges to a fixed point, $\bar{\mathbf{w}}^* = \langle(\bar{\mathbf{x}}^*)^T, (\bar{\Theta}^*)^T\rangle^T$. □

Proof of Theorem 10: Begin by assuming that $\bar{\mathbf{w}}^* = \langle(\bar{\mathbf{x}}^*)^T, (\bar{\Theta}^*)^T\rangle^T$ is not a local minimum. Then, there exists a feasible direction $\Delta\bar{\mathbf{w}} = \langle\Delta\bar{\mathbf{x}}^T, \Delta\bar{\Theta}^T\rangle^T$ from $\bar{\mathbf{w}}^*$ in the solution space such that

$$f(\bar{\mathbf{w}}^* + \lambda\Delta\bar{\mathbf{w}}) \leq f(\bar{\mathbf{w}}^*) \tag{42}$$

for some $\lambda \in (0, \infty)$ and

$$f(\bar{\mathbf{w}}^* + \sigma\Delta\bar{\mathbf{w}}) < f(\bar{\mathbf{w}}^*) \tag{43}$$

for some $\sigma \in [0, \lambda)$ where $f(\bar{\mathbf{w}})$ is the cost function. Let W be the set of all feasible directions at $\bar{\mathbf{w}}^*$. Let $\Delta\bar{\mathbf{w}} \in W$ where $\Delta\bar{\mathbf{w}} = \langle \Delta\bar{\mathbf{x}}^T, \Delta\bar{\Theta}^T \rangle^T$ and let $\bar{\mathbf{w}}_1 = \bar{\mathbf{w}}^* + \lambda\Delta\bar{\mathbf{w}} = \langle \bar{\mathbf{x}}_1^T, \bar{\Theta}_1^T \rangle^T$. Two mutually exclusive cases can occur: $\bar{\mathbf{x}}^*$ can be a unique optimal solution to (33) when $\bar{\Theta}$ is held constant, or $\bar{\mathbf{x}}^*$ can be a non-unique optimal solution. The first case occurs when $\bar{\mathbf{c}}(\bar{\Theta})$ supports the polyhedral set of feasible solutions only at the vertex, $\bar{\mathbf{x}}^*$. The second occurs when $\bar{\mathbf{c}}(\bar{\Theta})$ supports the set at vertex, $\bar{\mathbf{x}}$, and is also parallel to one of the faces of the polyhedral set which also contains $\bar{\mathbf{x}}^*$.

1. $\bar{\mathbf{x}}^*$ is a unique optimal solution.

   $\bar{\mathbf{x}}^*$ being optimal implies that for all feasible directions, $\Delta\bar{\mathbf{x}}$, from $\bar{\mathbf{x}}^*$ [1]

   $$\bar{\mathbf{c}}(\bar{\Theta}^*)^T \Delta\bar{\mathbf{x}} > 0.$$

   Moving in direction $\Delta\bar{\mathbf{w}} = \langle \Delta\bar{\mathbf{x}}^T, \Delta\bar{\Theta}^T \rangle^T$ changes $\bar{\Theta}^*$ as $\bar{\Theta}_1 = \bar{\Theta}^* + \lambda\Delta\bar{\Theta}$. This rotates the cost vector as $\bar{\mathbf{c}}(\bar{\Theta}_1) = \bar{\mathbf{c}}(\bar{\Theta}^*) + \lambda\hat{\mathbf{c}}(\Delta\bar{\Theta})$ where $\hat{\mathbf{c}}(\Delta\bar{\Theta})$ is a term that denotes the change in the cost vector due to $\Delta\bar{\Theta}$. If λ is small enough, then $\lambda\hat{\mathbf{c}}(\Delta\bar{\Theta})^T\Delta\bar{\mathbf{x}} < \bar{\mathbf{c}}(\bar{\Theta}^*)^T\Delta\bar{\mathbf{x}}$ and $\bar{\mathbf{c}}(\bar{\Theta}_1)^T\Delta\bar{\mathbf{x}} > 0$ for all feasible $\Delta\bar{\mathbf{x}}$ from $\bar{\mathbf{x}}^*$. Thus, $\bar{\mathbf{x}}^*$ is still the optimal solution to the LP problem associated with (33). If $\bar{\mathbf{x}}^*$ is the optimal solution, then

   $$\bar{\mathbf{c}}(\bar{\Theta}_1)^T\bar{\mathbf{x}}^* < \bar{\mathbf{c}}(\bar{\Theta}_1)^T\bar{\mathbf{x}}_1.$$

   The minimizing solution to the quadratic optimization of $\bar{\mathbf{c}}(\bar{\Theta})^T\bar{\mathbf{x}}^*$ with respect to $\bar{\Theta}$ is $\bar{\Theta}^*$. Thus,

   $$\bar{\mathbf{c}}(\bar{\Theta}^*)^T\bar{\mathbf{x}}^* < \bar{\mathbf{c}}(\bar{\Theta}_1)^T\bar{\mathbf{x}}^* < \bar{\mathbf{c}}(\bar{\Theta}_1)^T\bar{\mathbf{x}}_1$$

   For feasible direction $\Delta\bar{\mathbf{w}} = \langle \Delta\bar{\mathbf{x}}^T, \Delta\bar{\Theta}^T \rangle^T$, the cost of the solution $\bar{\mathbf{w}}^* = \langle (\bar{\mathbf{x}}^*)^T, (\bar{\Theta}^*)^T \rangle^T$ is less than the cost of the solution $\bar{\mathbf{w}}_1 = \langle \bar{\mathbf{x}}_1^T, \bar{\Theta}_1^T \rangle^T$. Since $\Delta\bar{\mathbf{w}}$ was any arbitrary feasible direction from $\bar{\mathbf{w}}^*$, there is no feasible direction that satisfies (42) and (43), which is a contradiction. Thus, the assumption is incorrect and $\bar{\mathbf{w}}^* = \langle (\bar{\mathbf{x}}^*)^T, (\bar{\Theta}^*)^T \rangle^T$ is a local minimum.

2. $\bar{\mathbf{x}}^*$ is a non-unique optimal solution to the LP problem.

   $\bar{\mathbf{x}}^*$ being non-unique implies that for all feasible directions $\Delta\bar{\mathbf{x}}$ from $\bar{\mathbf{x}}^*$

   $$\bar{\mathbf{c}}(\bar{\Theta}^*)^T\Delta\bar{\mathbf{x}} \geq 0$$

   This can be broken down into two cases, those for which $\bar{\mathbf{c}}(\bar{\Theta}^*)^T\Delta\bar{\mathbf{x}} > 0$ and those for which $\bar{\mathbf{c}}(\bar{\Theta}^*)^T\Delta\bar{\mathbf{x}} = 0$. The first considers those directions in which movement causes a strict increase in the cost function, while the second considers those directions in which movement causes no change in the cost function.

(a) Directions for which $\bar{c}(\bar{\Theta}^*)^T \Delta\bar{x} > 0$.

For movement in these directions, λ can be chosen small enough that $\lambda\hat{c}(\Delta\bar{\Theta})^T \Delta\bar{x} < \bar{c}(\bar{\Theta}^*)^T \Delta\bar{x}$ and the reasoning of case 1 applies, which generates the contradiction.

(b) Directions for which $\bar{c}(\bar{\Theta}^*)^T \Delta\bar{x} = 0$.

Here, the cost hyperplane $\bar{c}(\bar{\Theta}^*)$ is parallel to one of the constraint planes a_i where a_i is a row of A. This implies a set of optimal solutions $\{\bar{x} : \bar{x} = \beta\bar{x}^* + (1-\beta)\bar{x}_1^*, \beta \in [0,1]\}$ where $\bar{x}^*$ and $\bar{x}_1^*$ are extreme points of the polyhedral set of feasible solutions and $\Delta\bar{x}_1 = (\bar{x}_1^* - \bar{x}^*)/\|\bar{x}_1^* - \bar{x}^*\|$. Thus, $\Delta\bar{x}_1$ is a feasible direction from $\bar{x}^*$ to $\bar{x}_1^*$ and $-\Delta\bar{x}_1$ is a feasible direction from $\bar{x}_1^*$ to $\bar{x}^*$. Since $\bar{x}_1^*$ is also an optimal solution, $\bar{c}(\bar{\Theta}^*)^T \Delta\bar{x} \geq 0$ for all feasible directions from $\bar{x}_1^*$ and $\bar{c}(\bar{\Theta}^*)^T(-\Delta\bar{x}_1) = 0$ for the feasible direction $-\Delta\bar{x}_1$ leading from $\bar{x}_1^*$ to $\bar{x}^*$. Without loss of generality, assume $\Delta\bar{x}_1$ is such that

$$\lambda\hat{c}(\Delta\bar{\Theta})^T \Delta\bar{x}_1 > 0$$

which implies that

$$\bar{c}(\bar{\Theta}_1)^T \Delta\bar{x}_1 = \bar{c}(\bar{\Theta}^*)^T \Delta\bar{x}_1 + \lambda\hat{c}(\Delta\bar{\Theta})^T \Delta\bar{x}_1 > 0$$

With this small rotation of the cost plane, $\bar{x}^*$ is the unique optimal solution to the LP problem. Thus, $\bar{c}(\bar{\Theta}_1)^T\bar{x}^* < \bar{c}(\bar{\Theta}_1)^T\bar{x}_1$. Since $\bar{\Theta}^*$ is the optimal quadratic minimizer for (33) subject to constant $\bar{x} = \bar{x}^*$, then $\bar{c}(\bar{\Theta}^*)^T\bar{x}^* < \bar{c}(\bar{\Theta}_1)^T\bar{x}^* < \bar{c}(\bar{\Theta}_1)^T\bar{x}_1$. This again leads to a contradiction which implies that the solution, $\bar{w}^*$ is a local minimum.

□

A.3 Computational property proofs

Proof of Theorem 11: First, the assumption that S is chosen as indicated guarantees that $\|\bar{w}^{(k+1)} - \bar{w}^{(k)}\|$ will be a Cauchy sequence for $k \geq K$. Since $\|\bar{w}^{(k+1)} - \bar{w}^{(k)}\|$ is Cauchy, the norm of the difference between the current and optimal solutions can be expressed in closed form as

$$\|\bar{w}^{(k)} - \bar{w}^*\| < 2(\delta(1+\nu)+\nu)(1+MK)\frac{\Delta^{(k)}}{1-\beta} \tag{44}$$

Using (44), examine how long it will take for $\|\bar{w}^{(k)} - \bar{w}^*\| < \epsilon$. The goal is to find the smallest k that violates

$$\epsilon < 2(\delta(1+\nu)+\nu)(1+MK)\frac{\Delta^{(k)}}{1-\beta}$$

By definition of the duality gap, $\Delta^{(k)} = \beta^k \Delta^{(0)} < \beta^k (1/\epsilon)$. So

$$\epsilon < 2(\delta(1+\nu)+\nu)(1+MK)\frac{\beta^k(1/\epsilon)}{1-\beta}.$$

Taking the base 2 logarithm,

$$\log_2 \epsilon < \log_2 2(\delta(1+\nu)+\nu)(1+MK) + \log_2 1/\epsilon + \log_2 \frac{1}{1-\beta} + k \log_2 \beta$$

After some arithmetic, one finds that

$$k < \left(2\log_2 1/\epsilon + \log_2 2(\delta(1+\nu)+\nu)(1+MK) + \log_2 \frac{1}{1-\beta}\right) / \log_2 \frac{1}{\beta}$$

But $\log_2(1+\mu) \geq \mu/(1+\mu)$ [5], and using this expression for $\log_2(1/\beta)$ results in

$$k < \left(2\log_2 1/\epsilon + \log_2 2(\delta(1+\nu)+\nu)(1+MK) + \log_2 \frac{1}{1-\beta}\right)(\sqrt{n}/\nu + 1)$$

Also, $\log_2(1/(1-\beta)) = \log_2(\sqrt{n}/\nu + 1)$, so

$$k < (2\log_2 1/\epsilon + \log_2 2(\delta(1+\nu)+\nu)(1+MK) + \log_2(\sqrt{n}/\nu + 1))(\sqrt{n}/\nu + 1)$$

Since $2(\delta(1+\nu)+\nu)(1+MK)$ and ν are constants, $k = O(\sqrt{n}\log_2(\sqrt{n}/\epsilon))$ iterations. □

References

[1] Bazaraa, M.S., Sherali, H.D., and Shetty, C.M. (1993). **Nonlinear Programming: Theory and Algorithms**. John Wiley and Sons, Inc.

[2] Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals, and Systems*, **2**, 304-314

[3] Dempster, A.P., Laird, N.M., and Rubin, D.B., (1977). Maximum Likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society.* **39**, 1-38.

[4] Golub, G.H., Van Loan, C.F.(1989). **Matrix Computations**. The Johns Hopkins University Press, Baltimore, Maryland.

[5] Gonzaga, C.C. (1992). Path following methods for linear programming. *SIAM Review.* **34**, 167-224.

[6] Hadamard, J. (1964). **La Theorie des Equations aux Derivees Partielles**. Edition Scientifiques, Beijing.

[7] Hornik, K. M. Stinchcombe, and H. White (1989). Multilayer feedfoward networks are universal approximators, *Neural Networks.* **2**, 359-366.

[8] Jacobs, R.A., Jordan, M.I., Nowlan, S.J., and Hinton, G.E. (1991). Adaptive mixtures of local experts. *Neural Computation.* **3**, 79-87.

[9] Jordan, M.I., Xu, L. (1993). Convergence results for the EM approach to mixtures of experts architectures. *Technical Report 9303.* MIT Computational Cognitive Science, September 1993.

[10] Karmarkar, N. (1984). a new polynomial-time algorithm for linear programming. *Combinatorics.* **4**, 373-395.

[11] Lemmon, M.D., Szymanski, P.T. (1994). Interior point implementations of alternating minimization training. In Tesauro, Touretzky, and Leen (Eds.) **Advances in Neural Information Processing Systems 7**,574-582, San Mateo California, MIT Press.

[12] Park, J. and Sandberg, I.W. (1991), Universal approximation using radial basis function networks. *Neural Computation.* **3**, 246-257.

[13] Poggio, T. , Girosi, F. (1990). Networks for Approximation and Learning. *Proceedings of the IEEE*, **78**, 1481-1497.

[14] Redner, R.A., Walker, H.F. (1984). mixture densities, maximum likelihood and the EM algorithm. *SIAM Review*, **26**, 195-239.

[15] Szymanski, P.T. (1995) *Alternating Minimization of Non-convex Constrained Optimizations via Interior Point Techniques.* Ph.D. thesis, Dept. of Electrical Engineering, University of Notre Dame, November 1995.

APPENDIX B

C.J. Bett and M.D. Lemmon
"Finite-Horizon Induced L-infinity Performance of Linear Parameter Varying Systems"
Released as Technical Report ISIS-96-005, University of Notre Dame
submitted for publication to *Automatica* under the title
"An Approach to Amplitude and Transient Control of Linear Parameter Varying Systems".

An Approach to Amplitude and Transient Control of Linear Parameter Varying Systems

Christopher J. Bett and Michael D. Lemmon*

Department of Electrical Engineering
University of Notre Dame
Notre Dame, IN 46556 USA

original version: January 1997
revised: November 1997

Abstract

This paper considers the problem of controlling continuous-time linear parameter varying (LPV) systems when performance is measured in terms of prespecified bounds on plant signal amplitude and transient decay rate. Such problems arise in tasks where signal amplitude is more important than signal energy and in switched controller problems where quantifying transient decay is essential. Sufficient conditions for the synthesis of constant gain state feedback controllers for bounded amplitude and transient performance of LPV systems are proven. These conditions are useful in that they can be reformulated as linear matrix inequality (LMI) conditions amenable to efficient numerical techniques. An illustrative numerical example is included.

*The authors gratefully acknowledge the partial financial support of the Army Research Office (DAAH04-95-1-0600, DAAH04-96-1-0134). Corresponding author: M.D. Lemmon, phone:(219)631-8309, email:lemmon@maddog.ee.nd.edu

1 Introduction

In any control system design problem, there is a performance objective which is to be achieved by the design. In the continuous-time $\mathcal{H}_\infty$ design paradigm, for example, performance is measured with respect to the induced $\mathcal{L}_2$ norm of the closed loop system which corresponds to the gain that the system applies to the *energy* of an input signal. Another measure of system performance is the ***peak value*** of an output signal, especially in the presence of input signals which have bounded peak value but unbounded energy. Such control problems are prevalent in the literature: motor control problems with electrical (voltage or current) or mechanical (motion) restrictions and process control problems with chemical concentration restrictions are two examples. In each of these examples, violation of the restrictions could lead to performance degradation and possibly catastrophic failure of the system. In these problems, therefore, it is more important to control the peak value of the plant signals than the energy contained in the signals. In recent years, significant effort has been directed towards the problem of designing control systems which guarantee such prescribed bounds on peak signal values.

For discrete-time systems, the problem of controlling signal amplitudes is known as the ℓ_1-optimal control problem. First stated in [24], this problem is to design a controller which minimizes the effects of persistent (bounded amplitude) disturbances as measured by the induced ℓ_∞-norm (the peak-to-peak gain). A solution to the linear MIMO ℓ_1-optimal control problem appeared in [12], where optimal controllers are synthesized by solving a linear programming problem. Results for the discrete-time case have been extended to handle system uncertainty [10] [22] [19]. As with the solution to the deterministic case, controller synthesis amounts to solving linear programming problems.

While there have been significant advances in the discrete-time problem, the continuous-time problem has not enjoyed the same success. For continuous-time systems, the problem of controlling signal amplitude is denoted the $\mathcal{L}_1$ or induced-$\mathcal{L}_\infty$ optimal control problem. In [11], it is shown that optimal solutions to this problem are irrational or infinite-dimensional, even for rational and finite-dimensional plants. Furthermore, solving the problem exactly involves solving a nonlinear programming problem and while approximate solutions, which are rational and finite dimensional, may be found via linear programming problems, the solutions are still of arbitrarily high order.

An alternative approach for the $\mathcal{L}_1$ optimal control problem was recently proposed in [16]. There, instead of synthesizing a controller to explicitly minimize the induced-$\mathcal{L}_\infty$ norm of the system, an upper bound on the induced-$\mathcal{L}_\infty$ norm is minimized. Although the solution is suboptimal and may be conservative [23], the

solution method involves only the solution of a Riccati matrix inequality. This, in turn, may be expressed as a linear matrix inequality (LMI) problem [8], for which computationally efficient algorithms are available. Furthermore, the synthesized controller is guaranteed to have order no higher than the plant order. The results in [16] concern linear, time-invariant systems only.

In this paper, the constant gain state feedback results presented in [16] are extended to deal with linear parameter varying (LPV) systems [20]. LPV systems are linear dynamical systems whose coefficient matrices depend on a time-varying parameter in a piecewise-continuous fashion. The parameter varies over some known compact subset of Euclidean space, called the parameter set. LPV systems have been studied extensively for $\mathcal{L}_2$ performance problems especially in the context of gain scheduling ([21],[17],[2],[1]). Under appropriate conditions, a self-scheduled (LPV) controller may be synthesized for the LPV system. These controllers have the advantage of increasing both performance and the allowable size of the parameter set relative to LTI controllers. The bounded-amplitude self-scheduled controller problem is addressed in [7].

While a parameter-dependent controller may exist, the problem of constructing LTI controllers for LPV systems is still important because it is not always possible to implement a self-scheduled controller due to hardware constraints. In other words, it may only be possible to implement a small bank of controller gains. If the controller gains are *switched* according to some scheduling rule, then transient behavior becomes critical to bounded amplitude performance. These ideas are seen in [14] and in the the early works [5][4], where one is concerned with ensuring that the state trajectory enter a target set within a prescribed time interval.

The remainder of this paper is organized as follows. Section 2 formally defines the notion of LPV systems, parameter variations and associated performance measures used in this paper. (Section 2 also presents the notation used throughout the paper.) Section 3 presents the bounded amplitude performance results related to a class of LPV systems. This section contains the main results of the paper (theorem 3.1) which states sufficient conditions for bounded amplitude performance of LPV systems. This result is proven and discussed with respect to practical implementation of the conditions. The section concludes with a reformulation of the condition as an LMI (theorem 3.2). Section 4 presents a numerical example to illustrate the theoretical results.

Many of the definitions for LPV systems presented in this section are generalized from [3]; the induced-$\mathcal{L}_\infty$ norm results are generalizations of the LTI results reported in [16].

2 Mathematical Preliminaries

The section begins by formally defining a linear parameter varying (LPV) system and discussing various notions of parameter variation, specifically the notion of finite horizon behavior. The performance measure considered in this section, the induced-$\mathcal{L}_\infty$ norm, is defined for a LPV system with a special emphasis on finite horizon behavior. Many of the definitions for LPV systems presented in this section are generalized from [3].

Definition 2.1 establishes the notation for $\mathcal{L}_\infty$ signal norms used throughout the paper.

Definition 2.1 *For a finite constant $T > 0$, the finite-horizon infinity norm of a signal $f : \mathbb{R}^+ \to \mathbb{R}^n$ is defined as*

$$\|f\|_{\infty,[0,T]} := \operatorname{ess\,sup}_{t\in[0,T]} \|f(t)\|$$

where $\|\cdot\|$ denotes the Euclidean l_2 vector norm. The linear space $\mathcal{L}^n_\infty[0,T]$ is defined by

$$\mathcal{L}^n_\infty[0,T] := \left\{ f : \mathbb{R}^+ \to \mathbb{R}^n \,\middle|\, \|f\|_{\infty,[0,T]} < \infty \right\}$$

The subset $\left\{ f : \mathbb{R}^+ \to \mathbb{R}^n \,\middle|\, \|f\|_{\infty,[0,T]} \le 1 \right\} \subset \mathcal{L}^n_\infty[0,T]$ is denoted $\mathcal{BL}^n_\infty[0,T]$.

The infinite-horizon infinity norm of a signal $f : \mathbb{R}^+ \to \mathbb{R}^n$ is defined as

$$\|f\|_{\infty,[0,\infty)} := \operatorname{ess\,sup}_{t\in[0,\infty)} \|f(t)\|$$

where $\|\cdot\|$ denotes the Euclidean l_2 vector norm. The linear space $\mathcal{L}^n_\infty[0,\infty)$ is defined by

$$\mathcal{L}^n_\infty[0,\infty) := \left\{ f : \mathbb{R}^+ \to \mathbb{R}^n \,\middle|\, \|f\|_{\infty,[0,\infty)} < \infty \right\}$$

The subset $\left\{ f : \mathbb{R}^+ \to \mathbb{R}^n \,\middle|\, \|f\|_{\infty,[0,\infty)} \le 1 \right\} \subset \mathcal{L}^n_\infty[0,\infty)$ is denoted $\mathcal{BL}^n_\infty[0,\infty)$. The spaces $\mathcal{L}^n_\infty[0,\infty)$ and $\mathcal{BL}^n_\infty[0,\infty)$ will often be denoted, respectively, $\mathcal{L}^n_\infty$ and $\mathcal{BL}^n_\infty$.

Definitions 2.2-2.3 formally establish the notion of a linear parameter varying system used in this section.

Definition 2.2 (Parameter Variation Set) *Given a compact subset $\Theta \subset \mathbb{R}^s$, the* parameter variation set *$\mathcal{F}_\Theta$ denotes the set of all piecewise continuous functions mapping $\mathbb{R}^+$ into Θ.*

The notation $\theta \in \mathcal{F}_\Theta$ denotes a function in the parameter variation set; $\theta \in \Theta$ denotes a vector in a compact subset of $\mathbb{R}^s$.

Note that both $\mathcal{F}_\Theta$ and, for instance, $\mathcal{L}^n_\infty$ represent signal spaces. Technically, it can easily be argued that $\mathcal{F}_\Theta \subset \mathcal{L}^s_\infty$ since $\mathcal{F}_\Theta$ consists of supremum bounded s-dimensional vectors which vary continuously in time.

In this paper, the following convention is followed: $\mathcal{F}_\Theta$ will always refer to parameter signals or parameter variations; $\mathcal{L}_\infty^n$ will refer to signals in the plant input, output or state space.

Definition 2.3 (Linear Parameter Varying (LPV) System) *Given a compact set $\Theta \subset \mathbb{R}^s$, and continuous functions $A : \mathbb{R}^s \to \mathbb{R}^{n\times n}$, $B : \mathbb{R}^s \to \mathbb{R}^{n\times n_w}$, $C : \mathbb{R}^s \to \mathbb{R}^{n_z\times n}$, and $D : \mathbb{R}^s \to \mathbb{R}^{n_z\times n_w}$, an n^{th} order linear parameter varying (LPV) system is a dynamical system whose dynamics evolve as*

$$\begin{bmatrix} \dot{x}(t) \\ z(t) \end{bmatrix} = \begin{bmatrix} A(\theta(t)) & B(\theta(t)) \\ C(\theta(t)) & D(\theta(t)) \end{bmatrix} \begin{bmatrix} x(t) \\ w(t) \end{bmatrix} \tag{1}$$

where $\theta \in \mathcal{F}_\Theta$.

3 Bounded Amplitude Performance Conditions

3.1 Theoretical Results

In many applications, it is necessary to consider performance of a system under amplitude constraints when the initial state of the system violates these constraints. In these cases, we must consider transient system behavior as well as performance after transients have decayed.

In this section, LPV systems are considered which take the form

$$\left[\begin{array}{c} \dot{x}(t) \\ z(t) \end{array}\right] = \left[\begin{array}{c|cc} A(\theta(t)) & B_w(\theta(t)) & B_u(\theta(t)) \\ \hline C_z(\theta(t)) & 0 & D_{zu}(\theta(t)) \end{array}\right] \begin{bmatrix} x(t) \\ w(t) \\ u(t) \end{bmatrix} \tag{2}$$

where $\theta \in \mathcal{F}_\Theta$. Here, $A : \mathbb{R}^s \to \mathbb{R}^{n\times n}$, $B_u : \mathbb{R}^s \to \mathbb{R}^{n\times n_u}$, $B_w : \mathbb{R}^s \to \mathbb{R}^{n\times n_w}$, $C_z : \mathbb{R}^s \to \mathbb{R}^{n_z\times n}$ and $D_{zu} : \mathbb{R}^s \to \mathbb{R}^{n_z\times n_u}$ are continuous mappings.

Lemma 3.1 *Given constant $\gamma > 0$, a positive definite matrix $Q \in \mathbb{R}^{n\times n}$, a matrix mapping $C : \mathbb{R}^s \to \mathbb{R}^{n_z\times n}$ and a parameter set $\Theta \subseteq \mathbb{R}^s$, then $x'C(\theta)'C(\theta)x \leq \gamma^2$ for all x such that $x'Q^{-1}x \leq 1$ and all $\theta \in \Theta$ if and only if $\gamma^2 Q^{-1} \geq C(\theta)'C(\theta)$ for all $\theta \in \Theta$.*

Lemma 3.1 relates the performance level, γ, to the size of a parameter varying (output) matrix, $C(\theta)$, through a positive definite matrix Q. The following theorem applies this result to specify conditions for finite-horizon bounded amplitude performance of the class of LPV systems characterized by (2).

Theorem 3.1 (Finite Horizon Performance) *Given scalar $\gamma > 0$ and the LPV system, $\Sigma(\Theta, A, B, C, D)$ with*

$$B(\theta) = \begin{bmatrix} B_w(\theta) & B_u(\theta) \end{bmatrix}, \; C_z(\theta) \text{ and } D(\theta) = \begin{bmatrix} 0 & D_{zu}(\theta) \end{bmatrix}$$

where $B_u : \mathbb{R}^s \to \mathbb{R}^{n \times n_u}$, $B_w : \mathbb{R}^s \to \mathbb{R}^{n \times n_w}$, $C_z : \mathbb{R}^s \to \mathbb{R}^{n_z \times n}$ and $D_{zu} : \mathbb{R}^s \to \mathbb{R}^{n_z \times n_u}$ are continuous mappings. Let Θ be a compact subset of $\mathbb{R}^s$. If there exist constants $\alpha > 0$ and $\beta \geq 0$, a positive definite matrix $Q \in \mathbb{R}^{n \times n}$ and a real matrix $V \in \mathbb{R}^{n_u \times n}$ satisfying

$$\begin{bmatrix} Q & QC_z'(\theta) + V'D_{zu}'(\theta) \\ C_z(\theta)Q + D_{zu}(\theta)V & \gamma^2 I \end{bmatrix} \geq 0 \tag{3}$$

and

$$QA'(\theta) + A(\theta)Q + (\alpha + \beta)Q + \frac{1}{\alpha}B_w(\theta)B_w'(\theta) + B_u(\theta)V + V'B_u'(\theta) \leq 0 \tag{4}$$

for all $\theta \in \Theta$, then the closed loop system under control $u = Kx$, where $K = VQ^{-1}$, will satisfy the following for $\theta \in \mathcal{F}_\Theta$:

1. *the conditions $w \in \mathcal{BL}_\infty^{n_w}$ and $x'(0)Q^{-1}x(0) \leq 1$ imply that*

$$x'(t)Q^{-1}x(t) \leq 1 \text{ and } z'(t)z(t) \leq \gamma^2 \text{ for all } t \geq 0, \tag{5}$$

2. *if $\beta > 0$, $w \in \mathcal{BL}_\infty^{n_w}$ and $x'(0)Q^{-1}x(0) = r_0 > 1$ then*

$$x'(t)Q^{-1}x(t) \leq 1 \text{ for all } t \geq t_d$$

where

$$t_d := -\frac{1}{\beta}\log\left(\frac{1}{r_0}\right)$$

The first part of the theorem is proven by demonstrating the invariance of an ellipsoid, $\{\xi \mid \xi'Q^{-1}\xi \leq 1\}$ to all allowable input disturbances and under all possible parameter variations. This is accomplished by establishing that a function $V(\xi) := \xi'Q^{-1}\xi$ is non-increasing outside of the ellipsoid. Lemma 3.1 is used to show that all system states contained in the ellipsoid map to performance satisfying outputs. The second part of the theorem is proven by bounding the decay rate of $V(\xi)$ when $\beta > 0$. A formal proof is located in the appendix.

Theorem 3.1 and its proof are a special case of a more general results on uniform ultimate boundedness for the control of uncertain systems [9] . The usefulness of this result, versus the more general one, is

that theorem 3.1 is more amenable to computational methods which allow for the automated synthesis of controllers. This is accomplished by exploiting the underlying linear structure of the system.

Theorem 3.1 demonstrates that sufficient conditions for bounded amplitude performance of a class of LPV systems may be characterized by a Riccati matrix inequality (equation 4) coupled with a constraint on the size of the output matrix (equation 3). The theorem indicates that the initial state of the LPV system is an important condition for bounded amplitude performance. In previous work [16], the initial state of the system is assumed to be zero, so that transient effects are neglected. In part 2 of the theorem, it is seen that under appropriate conditions, the system state can be guaranteed to decay at a specific rate, β. The bound, t_d, on the decay time will be called the *dwell-time*. This definition is consistent with notions established in the switching control literature (e.g., [15][18][6]) where a controller is required to "dwell" in the feedback loop long enough for switching transients to decay.

The constants α and β which appear in the conditions of the theorem have a physical interpretation as well; they provide an indication of the speed of the system and the ability of the system to reject disturbances which enter through the matrix $B_w(\theta)$. Consider, for a fixed $\theta \in \Theta$ inequality 4, rewritten as

$$Q^{1/2}(A_{cl}(\theta) + \frac{\alpha+\beta}{2}I)'Q^{-1/2} + Q^{-1/2}(A_{cl}(\theta) + \frac{\alpha+\beta}{2}I)Q^{1/2} + \frac{1}{\alpha}Q^{-1/2}B_w(\theta)B_w'(\theta)Q^{-1/2} \leq 0 \quad (6)$$

where $A_{cl}(\theta) := A(\theta) + B_u(\theta)K$. It is clear from eigenvalue perturbation theory that if

$$\alpha + \beta > -2 \max \mathrm{Re}(\mathrm{eig}(A_{cl}(\theta))). \quad (7)$$

then $A_{cl}(\theta) + \frac{1}{2}(\alpha+\beta)I$ will have at least one eigenvalue in the open right half-plane. This implies that the matrix

$$Q^{1/2}(A_{cl}(\theta) + \frac{\alpha+\beta}{2}I)'Q^{-1/2} + Q^{-1/2}(A_{cl}(\theta) + \frac{\alpha+\beta}{2}I)Q^{1/2}$$

will be indefinite for any positive definite matrix Q. Since $\frac{1}{\alpha}Q^{-1/2}B_w(\theta)B_w'(\theta)Q^{-1/2} \geq 0$ for any $\alpha > 0$, the inequality in equation 6 can have no positive definite solution Q if equation 7 is satisfied. The constants α and β are thus related to the slowest eigenvalue of the matrix $A_{cl}(\theta)$. (This observation is similar to that made in [16].) Since this must be true for any $\theta \in \Theta$, α and β provide an indication of the speed of the LPV system. Note that the notion of eigenvalues for time-varying systems is not well-defined, so this argument only makes sense for slow parameter variations. However, the argument lends intuition and agrees with the final part of theorem 3.1 where $\beta > 0$ yields an exponential bounding function on the trajectory of the system.

The statement of the theorem implies that for some values of $\alpha > 0$, a controller may not exist which satisfies the performance conditions. This may be seen by interpreting α as a reflection of the sensitivity of the system to exogenous disturbances. Note that as α decreases, the term $\frac{1}{\alpha}Q^{-1/2}B_w(\theta)B_w'(\theta)Q^{-1/2}$ becomes larger, making inequality 6 more difficult to solve. On the other hand, a larger α diminishes the influence of this term. Combining this observation with the above argument is consistent with the usual notion that a faster system possesses better disturbance rejection properties.

Remark: Note that characterizing the performance of open loop systems ($u = 0$) is a special case of theorem 3.1. In this instance, the conditions are rewritten with $V = 0$.

3.2 Finding Solutions

Theorem 3.1 presents a set of simple conditions for LPV system performance, but two important computational issues must be addressed.

First, consider the matrix inequalities of (3) and (4) and note that the matrix inequality constraints are linear in V and γ^2, but *bilinear* in α, β and Q. (The $1/\alpha$ term in (4) may be transformed using Schur complements.) However, if one fixes α and β (which are scalars), then (3) and (4) become affine matrix inequality constraints in γ^2, V and Q, and hence amenable to efficient and commercially available optimization tools. Since α and β are related to the response of the closed loop system, it stands to reason that (3) and (4) might be solved to minimize γ^2 for a family of pairs (α, β), each time generating a new controller. For (α, β) pairs with a fixed β, this procedure generates a set of controllers which minimize an upper bound on amplitude performance, γ, with a guaranteed transient decay rate, β. A design engineer could generate a set of level curves from such data and choose a controller which best represents the desired performance objectives. An example of this idea is provided in section 4. We point out that α and β are scalars for problems of any dimension, so that number of optimization problems which must be solved to generate the level curves does not grow with the dimension of the system or the dimension of the parameter space.

Because the conditions of theorem 3.1 are sufficient for controller synthesis, a controller generated for a fixed pair (α, β) using the above procedure is guaranteed to achieve the designed amplitude and transient decay performance characteristics. However, because the conditions are not necessary, the controller may actually achieve higher performance than intended. To help quantify the difference between designed and achieved performance for a fixed controller, K, one may apply the above procedure on the *closed-loop* LPV

system dynamics with $V = 0$. This procedure could be used to generate an additional set of level curves which the design engineer could use to predict transient and amplitude performance of the closed-loop system. An example of this is provided in section 4.

A similar approach is outlined in [16] for the LTI problem with $\beta = 0$. In that paper, it is shown that for the closed-loop system, the minimum γ^2, subject to the matrix inequality constraints, is a convex function of α. While this property is not proven here for the more general case, experimental results imply convexity for constant values of β under parameter variations. Additional experimental results also indicate such a convex relationship for the controller synthesis problem, suggesting that an "optimal" controller can be found from the level curves.

The second issue is due to that fact that the matrix inequalities (3) and (4) depend continuously on the parameters $\theta \in \Theta$. This implies an infinite number of constraints which must be satisfied in order to apply the conditions of the theorem. Typically, one must resort to gridding the set Θ and solving a set of matrix inequalities simultaneously at all of the grid points, refining the grid until a satisfactory approximation is found. While applied in practice, such an approach becomes rapidly inefficient as the dimension of Θ grows.

By placing (practical) restrictions on the problem, the parameter dependence issue becomes more tractable. If the elements of the parameter vector are all amplitude bounded, i.e. $|\theta_i(t)| \leq 1$, $i = 1, \ldots, s$, then Θ is a polytope. If, in addition, the parameter dependence in the state-space matrices is assumed to be linear fractional, then one may find a parameter independent sufficient condition for a solution to (3) and (4). This condition is stated in the following theorem.

Theorem 3.2 *Given a compact set*

$$\Theta := \left\{ \theta \,\middle|\, \sup_{1 \leq i \leq s} |\theta_i| \leq 1 \right\} \subset \mathbb{R}^s$$

Let $\Delta(\theta) := diag(\theta_1 I_{r_1}, \ldots, \theta_s I_{r_s})$ *where* I_{r_i} *denotes the* $r_i \times r_i$ *identity matrix with* $r := \sum r_i$ *for* $i = 1, \ldots, s$. *Suppose that the elements of the state-space matrices in (2) are rational functions of the elements of* θ *so that*

$$\begin{bmatrix} A(\theta(t)) & B_w(\theta(t)) & B_u(\theta(t)) \\ C_z(\theta(t)) & 0 & D_{zu}(\theta(t)) \end{bmatrix} = \begin{bmatrix} A_s & B_w & B_u \\ C_z & 0 & D_{zu} \end{bmatrix} + \begin{bmatrix} B_p \\ C_p \end{bmatrix} (I - \Delta(\theta) D_{qp})^{-1} \Delta(\theta) \begin{bmatrix} C_q & D_{qw} & D_{qu} \end{bmatrix} \quad (8)$$

for constant matrices $A_s \in \mathbb{R}^{n\times n}$, $B_w \in \mathbb{R}^{n\times n_w}$, $B_u \in \mathbb{R}^{n\times n_u}$, $C_z \in \mathbb{R}^{n_z\times n}$, $D_{zu} \in \mathbb{R}^{n_z\times n_u}$, $B_p \in \mathbb{R}^{n\times r}$, $C_p \in \mathbb{R}^{n_z\times r}$, $C_q \in \mathbb{R}^{r\times n}$, $D_{qp} \in \mathbb{R}^{r\times r}$, $D_{qw} \in \mathbb{R}^{r\times n_w}$ *and* $D_{qu} \in \mathbb{R}^{r\times n_u}$.

If there exist constants $\alpha > 0$ *and* $\beta \geq 0$, *a positive definite matrix* $Q \in \mathbb{R}^{n\times n}$, *a matrix* $V \in \mathbb{R}^{n_u\times n}$ *and diagonal positive semi-definite matrices* Λ, Π *and* Ψ *such that*

$$\begin{bmatrix} Q & QC_z' + V'D_{zu}' & QC_p' + V'D_{qu}' \\ C_zQ + D_{zu}V & \gamma^2 I - C_q\Psi C_q' & -C_q\Psi D_{qp}' \\ C_qQ + D_{qu}V & -D_{qp}\Psi C_q' & \Psi - D_{qp}\Psi D_{qp}' \end{bmatrix} \geq 0 \qquad (9)$$

$$\begin{bmatrix} M_{11} & QC_p' + V'D_{qu}' + B_p\Lambda D_{qp}' & B_w & B_p \\ D_{qu}V + C_qQ + D_{qp}\Lambda B_p' & D_{qp}\Lambda D_{qp}' - \Lambda & 0 & 0 \\ B_w' & 0 & D_{qw}'\Pi D_{qw} - \alpha I & D_{qw}'\Pi D_{qp} \\ B_p' & 0 & D_{qp}'\Pi D_{qw} & D_{qp}'\Pi D_{qp} - \Pi \end{bmatrix} \leq 0 \qquad (10)$$

where

$$M_{11} = QA_s' + A_sQ + (\alpha+\beta)Q + V'B_u' + B_uV + B_p\Lambda B_p'$$

then for all $\theta \in \Theta$,

$$\begin{bmatrix} Q & QC_z'(\theta) + V'D_{zu}'(\theta) \\ C_z(\theta)Q + D_{zu}(\theta)V & \gamma^2 I \end{bmatrix} \geq 0$$

$$QA'(\theta) + A(\theta)Q + (\alpha+\beta)Q + \frac{1}{\alpha}B_w(\theta)B_w'(\theta) + B_u(\theta)V + V'B_u'(\theta) \leq 0$$

The preceding result is derived by applying the $\mathcal{S}$-procedure[26] in a straightforward manner typical of the approaches described in [8]. The theorem handles the case when the parameter variations are further restricted to being multi-affine. In this case, however, the $\mathcal{S}$-procedure can result in matrix inequalities that are overly restrictive. For multi-affine parameter dependence, however, it is sufficient to solve (3) and (4) simultaneously on the vertices of the polytope Θ.

4 Example

In this section, the theoretical results of the previous section are used to design a controller for a nonlinear process control model. The plant dynamics are given by

$$\begin{aligned} \dot{x}_{p1} &= -x_{p1} + u_1 \\ \dot{x}_{p2} &= -x_{p2} + (1 + x_{p1}^2)u_2 \end{aligned} \qquad (11)$$

The variables $x_{p1} \in \mathbb{R}$ and $x_{p2} \in \mathbb{R}$ represent the state of the plant; $u_1 \in \mathbb{R}$ and $u_2 \in \mathbb{R}$ are the control inputs. The plant is to be cycled through various operating points in the plant state space. For the illustrative purposes of this example, the control objective under consideration is to move the plant from an initial state near the operating point $(x_{p1}, x_{p2}) = (2.5, 2)$ to a point near $(x_{p1}, x_{p2}) = (1, 3)$ in 1 second according to the reference trajectory generated by the following dynamical reference model by

$$\begin{aligned} \dot{x}_{m1} &= x_{m1} - 1.63 - x_{m1}/(1 + 0.5 \sin 10(x_{m1} - 1.63)) \\ \dot{x}_{m2} &= 1 \end{aligned} \tag{12}$$

This reference model state trajectory is depicted in figure 1.

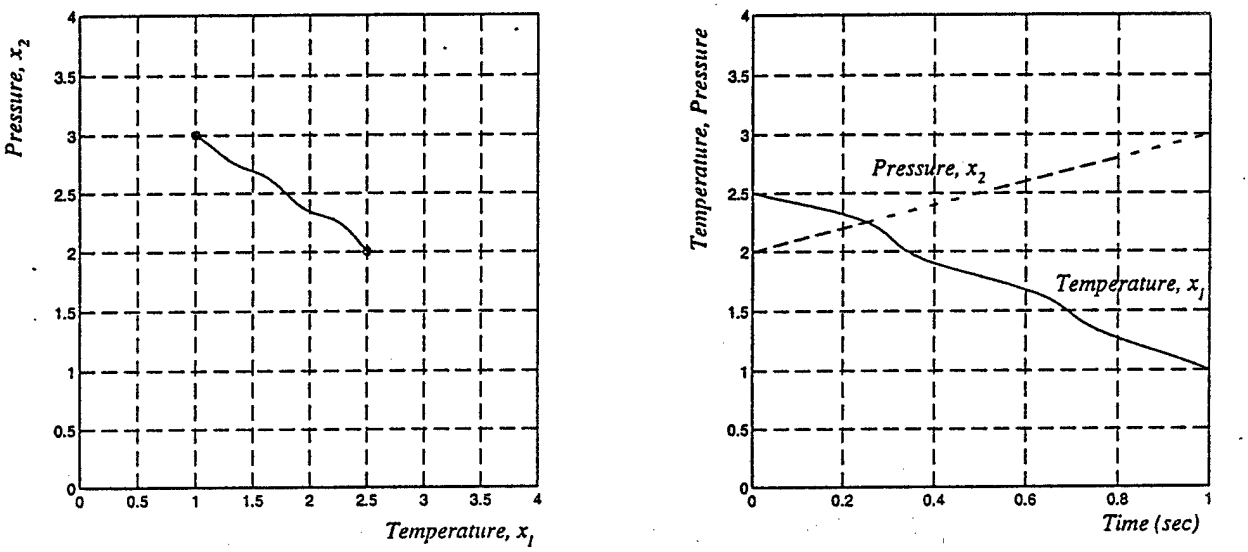


Figure 1: Reference trajectory for process control example

Defining the state error, $x := x_m - x_p$, the control objective is to synthesize an affine control

$$u = Kx + \tilde{u} \tag{13}$$

so that the performance output variable $z = C_1 x + D_{12} u$ satisfies the amplitude bound

$$\sup_{t \in [0,1]} \|z(t)\| \leq \gamma \tag{14}$$

where the performance weights are chosen as $C_1 = I$ and $D_{12} = 0.1I$.

After linearizing the plant at points along the reference trajectory, state and control dependent terms

can be grouped into parameters

$$\begin{bmatrix} \hat{\theta}_1(t) \\ \hat{\theta}_2(t) \\ \hat{\theta}_3(t) \end{bmatrix} = \begin{bmatrix} x_{m1} - x_{m1}/(2+\sin 10(x_{m1}-1.63)) \\ x_{m2} + (2x_{m1}x_1 - x_1^2)u_2 \\ x_{m1}^2/3 \end{bmatrix}. \tag{15}$$

By choosing a nominal design point, $\tilde{\theta}$, corresponding to states near the reference trajectory and setting

$$\tilde{u} = \begin{bmatrix} \tilde{\theta}_1 - 1.63 \\ \frac{\tilde{\theta}_2+1}{\tilde{\theta}_3+1} \end{bmatrix}, \tag{16}$$

the control synthesis objective may be restated as finding a control $u - \tilde{u} = Kx$ so that the LPV system

$$\begin{bmatrix} \dot{x} \\ z \\ y \end{bmatrix} = \begin{bmatrix} A(\theta) & B_w(\tilde{\theta}) + B_w(\theta) & B_u(\tilde{\theta}) + B_u(\theta) \\ C_1 & 0 & D_{12} \\ I & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ w \\ u - \tilde{u} \end{bmatrix} \tag{17}$$

where

$$A(\theta) = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix}, \; B_u(\theta) = \begin{bmatrix} -1 & 0 \\ 0 & -\theta_3 - 1 \end{bmatrix} \text{ and } B_w(\theta) = \begin{bmatrix} \theta_1 - 1.63 \\ \theta_2 + 1 \end{bmatrix} + B_u(\theta)\tilde{u}. \tag{18}$$

satisfies the required performance criteria. Here, the new parameters $\theta(t)$ represent deviations of $\hat{\theta}(t)$ from $\tilde{\theta}$, i.e. $\theta(t) := \hat{\theta}(t) - \tilde{\theta}$, and $w = 1$ is introduced as a fictitious disturbance. We assume that parameter variations are restricted to the polytope

$$\Theta := \left\{ \theta \,\middle|\, \sup_{1 \le i \le s} |\theta_i| \le 1 \right\} \subset \mathbb{R}^s$$

The derived state space data was used in the LMI of theorem 3.2 to study the achievable performance of state feedback controllers. In accordance with the analysis procedure outlines at the beginning of section 3.2, β was fixed and the matrices Q, V, Λ and Π were found which corresponded to the minimum feasible value of γ subject to the LMI constraints of theorem 3.2 for various values of α. Figure 2 shows the guaranteed performance level for values of α and β. The LMI problems were solved using the MATLAB with LMI Control Toolbox.

For the purposes of this example, the $\beta = 0$ curve was used to choose a controller. For a desired performance level of $\gamma = 0.4$, the minimum value of α which guaranteed performance was read from figure 2 to be $\alpha = 19$. This choice is indicated on the plot. The corresponding controller was determined, from $K = VQ^{-1}$ to be

$$K = \begin{bmatrix} 26.99 & 0 \\ 0 & 18.42 \end{bmatrix}$$

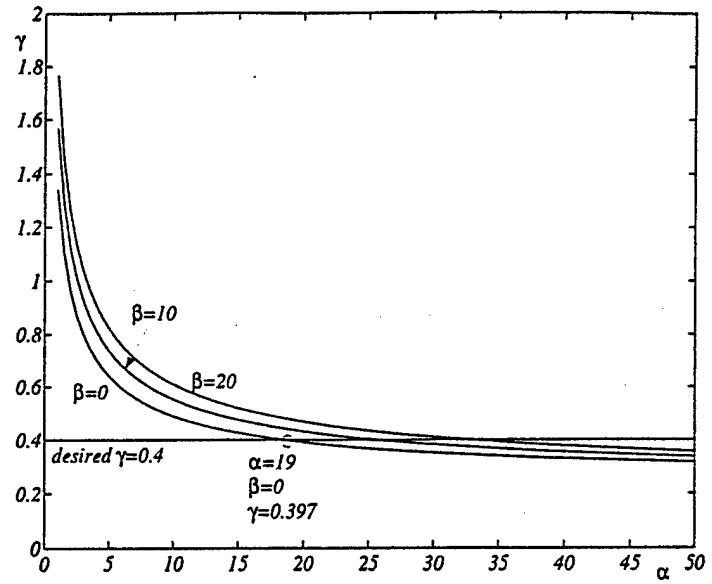


Figure 2: Guaranteed performance curves for open loop system.

As noted in section 3.2, while this controller is optimal with respect to the LMI synthesis constraints for the particular choice of $\alpha = 19$ and $\beta = 0$, the achieved performance may be tighter than indicated by the synthesis curves in figure 2. Figure 3 depicts the performance curves generated by solving the matrix inequalities with the closed-loop system data ($V = 0$). We remark that K found above is diagonal; this is a consequence of the plant modes of the problem being essentially decoupled. This has no significance for the nature of the results presented in the paper.

Two important quantities may be determined from the level curves of figure 3. The first is that the actual achieved amplitude bound is $\gamma = 0.364$; this is the bound which will be observed whenever the initial state satisfies the performance bounds. The second observation concerns initial states which violate the performance constraint $\gamma = 0.4$; in such cases, the norm of the performance variable will be bounded by a decaying exponential with rate $\beta \geq 5$. This second observation is made by noting that the performance curves for $\beta \leq 5$ dip below the $\gamma = 0.4$ level, while the curves for higher levels of β do not. Experiments with other designs show similar results. Note that the level curves are convex. Also note that the point corresponding to $(\alpha, \beta) = (19, 0)$ corresponds to $\gamma = 0.3950$ in both figures 2 and 3, as expected.

Finally, the controller was simulated in the closed loop system with performance violating initial conditions. The resulting performance variable norm, $\|z(t)\|$, is shown in figure 4 along with the bounding functions specified by theorem 3.1. The exponential decay rate used for the bounding exponential corresponded to $\beta = 5$. Note that once the performance level of $\gamma = 0.4$ is achieved, it is maintained for the

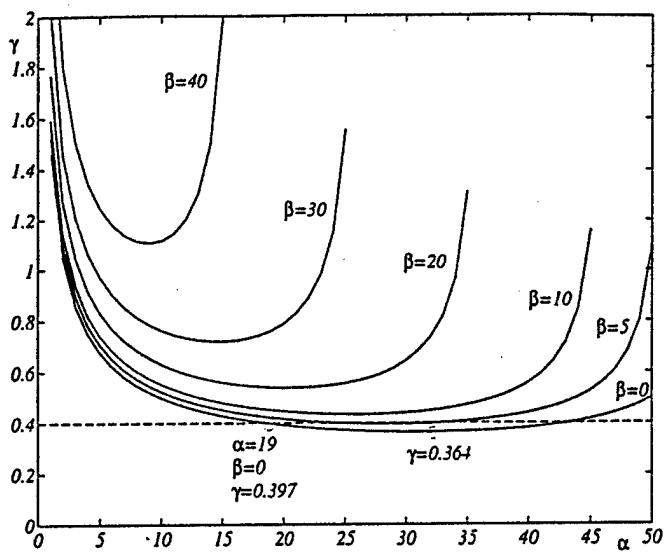


Figure 3: Achieved performance curves for closed loop system.

remainder of the trajectory.

5 Conclusions

This paper has presented an approach for synthesizing constant gain state feedback controllers for LPV systems which meet amplitude and transient decay performance constraints. Because the controllers are linear, time-invariant, the results pertain directly to scheduling problems where fixed gain controllers are *switched* into feedback with the plant according to some decision strategy. This is useful in problems where the structure of the controller is constrained, perhaps by physical limitations.

The approach presented in this paper is a generalization of the approach used for LTI systems in [16] where bounded amplitude controllers are synthesized by minimizing an upper bound on the induced-$\mathcal{L}_\infty$ of the closed-loop system. In [23] it was demonstrated that such an approach can be arbitrarily conservative with respect to the ℓ_1-induced norm. It should be possible to prevent this by placing additional (practical) linear which bound the condition number of the matrix Q of theorem 3.1.

The central theoretical result is theorem 3.1 which states sufficient conditions for the synthesis of bounded amplitude controllers guaranteeing a prescribed rate of transient decay as a matrix inequality feasibility problem. In many practical situations, this is extremely useful because there are now commercially available tools for solving linear matrix inequality problems. In these cases, the results provide a means for synthesis

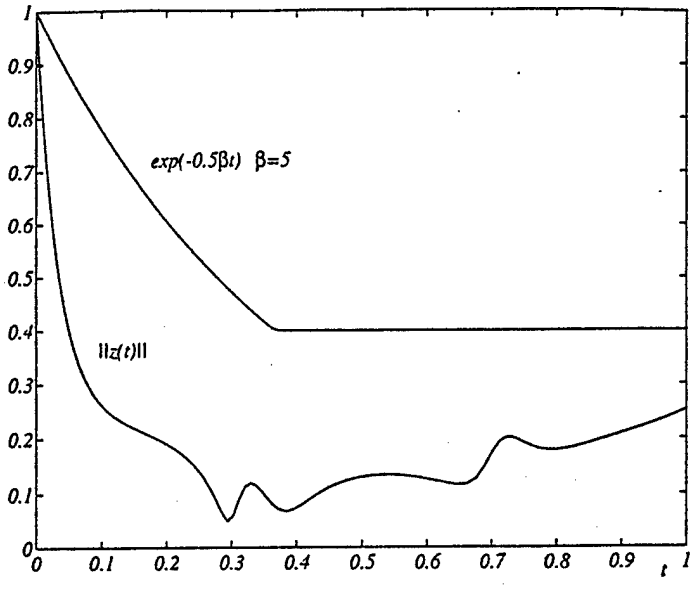


Figure 4: Simulated closed loop performance

of state feedback controllers and analysis of a general class of LPV systems.

The usefulness of theorem 3.1 is limited by the fact that it provides sufficient conditions only. Thus, if the matrix inequalities do not have a feasible point, no conclusion can be drawn regarding the existence of a robust controller. The conservativeness of the result might be alleviated by the introduction of a parameter-dependent $Q(\theta)$ [13][25]. in theorem 3.1. The tradeoff is that one must now make assumptions on the rate of parameter variation and to solve the matrix inequalities, one must resort to gridding the parameter set Θ. The results are also somewhat limited since the matrix inequalities are not linear in all of the design variables. However, this does not appear to present a problem from a computational standpoint. These limitations are alleviated by the systematic design and analysis approach presented in the paper.

Continued work along these lines involves extensions to output feedback controllers, self-scheduled controllers [7] and application of these results to switched agent control problems[14][6].

A Proofs of Results

Proof of Lemma 3.1: Let γ, Q, $C(\theta)$ and Θ be as defined in the lemma and suppose that

$$C'(\theta)C(\theta) \leq \gamma^2 Q^{-1} \quad \forall \theta \in \Theta \tag{19}$$

Since $Q > 0$, (19) is true if and only if

$$Q^{1/2}C'(\theta)C(\theta)Q^{1/2} \leq \gamma^2 I \quad \forall \theta \in \Theta \tag{20}$$

By the definition of an induced matrix norm, (20) is equivalent to

$$\left\| C(\theta)Q^{1/2} \right\|^2 \leq \gamma^2 \quad \forall \theta \in \Theta$$

or

$$\xi' Q^{1/2}C'(\theta)C(\theta)Q^{1/2}\xi \leq \gamma^2 \quad \forall \xi \in \{\xi \,|\, \xi'\xi \leq 1\} \text{ and } \forall \theta \in \Theta \tag{21}$$

Defining $x = Q^{1/2}\xi$, (21) is equivalent to

$$x'C'(\theta)C(\theta)x \leq \gamma^2 \quad \forall x \in \{x \,|\, x'Q^{-1}x \leq 1\} \text{ and } \forall \theta \in \Theta$$

completing the proof. □

Proof of theorem 3.1: Let $\gamma > 0$. Suppose that constants $\alpha > 0$ and $\beta \geq 0$, a positive definite matrix $Q \in \mathbb{R}^{n \times n}$ and a real matrix $V \in \mathbb{R}^{n_u \times n}$ are given which satisfy inequalities 3 and 4. Define $K = VQ^{-1}$. (K must exist since Q is positive definite.) First note from the theory of Schur complements that since $Q > 0$, inequality 3 is true if and only if

$$Q - \frac{1}{\gamma^2}Q(C_z(\theta) + D_{zu}(\theta)K)'(C_z(\theta) + D_{zu}(\theta)K)Q \geq 0$$

or, equivalently,

$$Q^{-1} \geq \frac{1}{\gamma^2}C_{cl}'(\theta)C_{cl}(\theta) \tag{22}$$

for all $\theta \in \Theta$ where $C_{cl}(\theta) := C_z(\theta) + D_{zu}(\theta)K$. Now, defining $A_{cl}(\theta) := A(\theta) + B_u(\theta)K$, inequality 4 may be rewritten as

$$QA_{cl}'(\theta) + A_{cl}(\theta)Q + (\alpha + \beta)Q + \frac{1}{\alpha}B_w(\theta)B_w'(\theta) \leq 0 \tag{23}$$

Defining $P = Q^{-1}$, the positive definiteness of Q implies that inequality 23 is true if and only if

$$A_{cl}'(\theta)P + PA_{cl}(\theta) + (\alpha + \beta)P + \frac{1}{\alpha}PB_w(\theta)B_w'(\theta)P \leq 0 \tag{24}$$

Clearly, for any $\theta \in \Theta$,

$$\frac{1}{\alpha}PB_w(\theta)B_w'(\theta)P \geq 0$$

so that if equation 4 holds for all $\theta \in \Theta$, then

$$A_{cl}'(\theta)P + PA_{cl}(\theta) + (\alpha + \beta)\,P \leq -\frac{1}{\alpha}PB_w(\theta)B_w'(\theta)P \leq 0$$

for all $\theta \in \Theta$. Thus, from the theory of Schur complements, equation 4 is true for all $\theta \in \Theta$ if and only if

$$\begin{bmatrix} A'_{cl}(\theta)P + PA_{cl}(\theta) + (\alpha+\beta)P & PB_w(\theta) \\ B'_w(\theta)P & -\alpha I \end{bmatrix} \leq 0 \quad \text{for all } \theta \in \Theta. \tag{25}$$

But equation 25 is satisfied if and only if for all $\theta \in \Theta$ and all $\xi \in \mathbb{R}^n$, $\nu \in \mathbb{R}^{n_w}$

$$\begin{bmatrix} \xi \\ \nu \end{bmatrix}' \begin{bmatrix} A'_{cl}(\theta)P + PA_{cl}(\theta) + (\alpha+\beta)P & PB_w(\theta) \\ B'_w(\theta)P & -\alpha I \end{bmatrix} \begin{bmatrix} \xi \\ \nu \end{bmatrix} \leq 0 \tag{26}$$

Equation 26 is true if and only if

$$\xi'(A'_{cl}(\theta)P + PA_{cl}(\theta) + \beta P)\xi + \nu' B'_w(\theta)P\xi + \xi' PB_w(\theta)\nu + \alpha(\xi' P\xi - 1) + \alpha(1 - \nu'\nu) \leq 0 \tag{27}$$

for all $\theta \in \Theta$ and all $\xi \in \mathbb{R}^n$, $\nu \in \mathbb{R}^{n_w}$. Equation 27, in turn, implies

$$\xi'(A'_{cl}(\theta)P + PA_{cl}(\theta))\xi + \nu' B'_w(\theta)P\xi + \xi' PB_w(\theta)\nu \leq -\beta\xi' P\xi \leq 0 \tag{28}$$

for all $\theta \in \Theta$ and all $\xi \in \mathbb{R}^n$ and $\nu \in \mathbb{R}^{n_w}$ such that

$$\xi' P\xi \geq 1 \text{ and } \nu'\nu \leq 1$$

To prove the first part of the theorem, consider any time $t \geq 0$ and any parameter variation $\theta \in \mathcal{F}_\Theta$. Now, define the function $V : \mathbb{R}^n \to \mathbb{R}$ by $V(\xi) := \xi' P\xi$. Along the trajectories of the closed loop LPV system, Σ_θ^{cl}, the time derivative of $V(x(t))$ is given by

$$\frac{d}{dt}V(x(t)) = x(t)'(A'_{cl}(\theta(t))P + PA_{cl}(\theta(t)))x(t) + w'(t)B'_w(\theta(t))Px(t) + x'(t)PB_w(\theta(t))w(t) \tag{29}$$

From the above argument, equation 4 and equation 28 imply

$$\frac{d}{dt}V(x(t)) \leq -\beta V(x(t)) \leq 0 \tag{30}$$

for any $x(t)$ and $w(t)$ satisfying $x(t)'Px(t) \geq 1$ and $w(t)'w(t) \leq 1$.

Now, suppose that for some $w \in \mathcal{BL}_\infty^{n_w}$ that there is a trajectory of Σ_θ^{cl} with initial state $x(0)$ satisfying $V(x(0)) = x'(0)Px(0) \leq 1$ and $V(x(t_f)) = x'(t_f)Px(t_f) > 1$ for some finite $t_f > 0$. Since $V(x(t))$ is differentiable in t, the Mean Value Theorem can be used to imply the existence of a time $\tau \in (0, t_f)$ for which $V(x(\tau)) = x'(\tau)Px(\tau) \geq 1$ and $\dot{V}(x(\tau)) > 0$. This is a contradiction of equation 30, so $x'(t)Px(t) = x(t)Q^{-1}x(t) \leq 1$ for all $t \geq 0$. By lemma 3.1 and equation 22, this is true if and only if

$$x'(t)C'_{cl}(\theta)C_{cl}(\theta)x(t) = z'(t)z(t) \leq \gamma^2$$

for all $t \geq 0$, proving the first part of the theorem.

To prove the second part of the theorem, assume that $\beta > 0$. For $\theta \in \mathcal{F}_\Theta$ and $w \in \mathcal{BL}_\infty^{n_w}$, equation 30 then implies that along any trajectory of Σ_θ for $t \geq 0$,

$$V(x(t)) \leq V(x(0)) + \int_0^t -\beta V(x(\tau))d\tau$$

so that from the Bellman-Gronwall Lemma,

$$V(x(t)) \leq V(x(0)) \exp(-\beta t) \tag{31}$$

Suppose that $V(x(0)) = x'(0)Px(0) = r_0 > 1$ and let

$$t_d := -\frac{1}{\beta} \log\left(\frac{1}{r_0}\right)$$

Then for all $t \geq t_d$, from equation 31

$$V(x(t)) \leq r_0 \exp(-\beta\left(-\frac{1}{\beta} \log\left(\frac{1}{r_0}\right)\right)) = 1$$

Equivalently, $x'(t)Px(t) \leq 1$ for all $t \geq t_d$. Substituting $Q^{-1} = P$ yields the final form of the result. □

References

[1] P. Apkarian and P. Gahinet. A convex characterization of gain-scheduled H_∞ controllers. *IEEE Transactions on Automatic Control*, 40(5):853–864, May 1995.

[2] G. Becker and A. Packard. Robust performance of linear parametrically varying systems using parametrically-dependent linear feedback. *Systems and Control Letters*, 23:205–215, 1994.

[3] Gregory Scott Becker. *Quadratic Stability and Performance of Linear Parameter Dependent Systems.* PhD thesis, University of California at Berkeley, 1993.

[4] D.P. Bertsekas. Infinite-time reachability of state-space regions by using feedback control. *IEEE Transactions on Automatic Control*, AC-17(5):604–613, October 1972.

[5] D.P. Bertsekas and I.B. Rhodes. On the minimax reachability of target sets and target tubes. *Automatica*, 7:233–247, 1971.

[6] C.J. Bett and M.D. Lemmon. Bounded amplitude performance of switched lpv systems with applications to hybrid systems. Technical Report ISIS-97-008, Department of Electrical Engineering, University of Notre Dame, Notre Dame, IN, 1997. Submitted to *Automatica*.

[7] C.J. Bett and M.D. Lemmon. Sufficient conditions for self-scheduled bounded amplitude control. Technical Report ISIS-97-009, Department of Electrical Engineering, University of Notre Dame, Notre Dame, IN, 1997. Submitted to *IEEE Trans. on Automatic Control.*

[8] Stephen Boyd, Laurent El Gaoui, Eric Feron, and Venkataramanan Balakrishnan. *Linear Matrix Inequalities in System and Control Theory.* Society for Industrial and Applied Mathematics, 1994.

[9] M.J. Corless and G. Leitmann. Continuous state feedback guaranteeing uniform ultimate boundedness for uncertain dynamic systems. *IEEE Transactions on Automatic Control*, AC-26(5):1139–1144, October 1981.

[10] M.A. Dahleh and I.J. Diaz-Bobillo. *Control of uncertain systems: a linear programming approach.* Prentice-Hall, 1995.

[11] M.A. Dahleh and J.B. Pearson. L^1-optimal compensators for continuous-time systems. *IEEE Transactions on Automatic Control*, 32(10):889–895, October 1987.

[12] M.A. Dahleh and J.B. Pearson. l^1-optimal feedback controllers for MIMO discrete-time systems. *IEEE Transactions on Automatic Control*, 32(4):314–322, April 1987.

[13] Edward W. Kamen and Pramod P. Khargonekar. On the control of linear systems whose coefficients are functions of parameters. *IEEE Transactions on Automatic Control*, AC-29(1):25–33, January 1984.

[14] M.D. Lemmon and C.J. Bett. Safe implementations of supervisory commands. *International Journal of Control*, to appear.

[15] A. S. Morse. Supervisory control of families of linear set-point controllers - part 1: Exact matching. *IEEE Transactions on Automatic Control*, 41(10):1413–1431, October 1996.

[16] K. Nagpal, J. Abedor, and K. Poolla. An LMI approach to peak-to-peak gain minimization: filtering and control. In *Proceedings of the American Control Conference*, pages 742–746, Baltimore, Maryland, June 1994.

[17] Andy Packard. Gain scheduling via linear fractional transformations. *Systems and Control Letters*, 22:79–92, 1994.

[18] Kameshwar Poolla and Jeff. S. Shamma. Optimal asymptotic robust performance via nonlinear controllers. *International Journal of Control*, 62(6):1367–1389, December 1995.

[19] Jeff S. Shamma. Optimization of the ℓ^∞-induced norm under full state feedback. *IEEE Transactions on Automatic Control*, 41(4):533–544, 1996.

[20] Jeff S. Shamma and Michael Athans. Guaranteed properties of gain scheduled control for linear parameter-varying plants. *Automatica*, 27(3):559–564, 1991.

[21] Jeff S. Shamma and Michael Athans. Gain scheduling: Potential hazards and possible remedies. *Control Systems Magazine*, 12(3):101–107, 1992.

[22] J.S. Shamma and K. Poolla. Optimal asymptotic robust performance via nonlinear controllers. *International Journal of Control*, 62(6):1367–1389, 1995.

[23] S.R. Venkatesh and M.A. Dahleh. Does star norm capture ℓ_1 norm? In *Proceedings of the American Control Conference*, pages 944–945, Seattle, WA, June 1995.

[24] M. Vidyasagar. Optimal rejection of persistent bounded disturbances. *IEEE Transactions on Automatic Control*, 31(6):527–535, June 1986.

[25] F. Wu, X.H. Yang, A. Packard, and G. Becker. Induced L_2-norm control for LPV systems with bounded parameter variation rates. In *Proceedings of the American Control Conference*, pages 2379–2383, Seattle, WA, June 1995.

[26] V.A. Yakubovich. The S-procedure in nonlinear control theory. *Vestnik Leningrad University Mathematics*, 4:73–93, 1977.

APPENDIX C

M.D. Lemmon and C.J. Bett
"Verification of Safe Inter-Event Behaviour in
Supervisory Hybrid Dynamical Systems"
presented to the Hybrid Systems and Autonomous Control
Workshop, Cornell University, Oct. 1996.
Released as Technical Report, ISIS-96-006, at the University of Notre Dame
to appear in *International Journal of Control* under the title
"Safe Implementations of Supervisory Commands".

Safe Implementations of Supervisory Commands

Revision of Paper submitted to
International Journal of Control
original submission:December, 1996
revision: August, 1997

Michael Lemmon and Christopher Bett
Department of Electrical Engineering
University of Notre Dame
Notre Dame, IN 46556

Interdisciplinary Studies of Intelligent Systems

Safe Implementations of Supervisory Commands

Michael Lemmon and Christopher J. Bett *
Department of Electrical Engineering
University of Notre Dame
Notre Dame, IN 46556

August 1997

Abstract

This paper compares two different types of control strategies used to safely implement supervisory commands of hybrid dynamical systems. Both approaches considered in this paper switch between members of a family of control agents to ensure that constraints on the plant state are not violated at any time. The first approach is motivated by a hybrid system architecture outlined in [Kohn 1993] and uses a Fliess functional series of the plant's output to form a system of linear inequalities characterizing safe control inputs. Control signals are determined by solving a sequence of linear programs. The second approach is a model reference control approach to hybrid systems introduced in [Lemmon 1996] and uses a known safe dynamical reference model to characterize the desired plant behavior. The controller is determined by representing the resulting error dynamics as a linear parameter varying system and applying linear robust control techniques to enforce a bounded amplitude performance level. The fundamental results underlying each of the methods are derived; both approaches are compared with regard to their complexity, performance, and sensitivity to modeling uncertainty. A numerical example is included for illustration.

1 Introduction

This paper considers the high level supervision of continuous time dynamical control systems evolving over a state set which is dense in $\Re^n$. It is assumed that a supervisory command is characterized by a set of guard conditions and a goal condition. These guard and goal conditions are inequality conditions on the plant's state. A control system

*The authors gratefully acknowledge the partial financial support of the Army Research Office (DAAH04-95-1-0600, DAAH04-96-1-0134).

is used to implement the supervisory command. This controller is said to be "safe" when the controlled plant's state trajectory triggers the goal condition in finite time without triggering any of the guard conditions. This paper compares two different types of controllers used to safely implement supervisory commands.

Both approaches considered in this paper switch between members of a family of control agents to ensure the guard conditions are not triggered. The first approach is motivated by a hybrid system architecture outlined in [Kohn 1993]. This approach uses a Fliess functional series of the plant's output to form a system of linear inequalities characterizing safe control inputs. In this method, control signals can be determined by solving a sequence of linear programs (LP). The second approach is a model reference control approach to hybrid systems introduced in [Lemmon 1996]. In this approach, the controlled plant follows a reference model which is known to be safe. The error dynamics of this system are represented as a linear parameter varying (LPV) system whose controllers enforce a bounded amplitude performance level. This paper formally derives the fundamental results behind both of these methods and compares both approaches with regard to their complexity, performance, and sensitivity to modeling uncertainty.

This paper is concerned with switched control systems as they appear in the design of hybrid dynamical systems. The primary contribution of this work concerns the formal development of two methods for the "safe" control of such systems. Safety is a *bounded amplitude* performance measure which seeks to ensure that the amplitude, $\max_t \|x(t)\|$, of a signal is appropriately bounded. For continuous-time systems there is very little work concerned with the control (switched or otherwise) of systems satisfying bounded amplitude performance measures. In particular, most of the prior work on switched dynamical systems has dealt with the assurance of induced $\mathcal{L}_2$ performance norms. In this regard, the results and methods of this paper provide a perspective on bounded

amplitude control which has not been well addressed in the academic community.

A formal definition of "safe" controllers is given in section 2. The remainder of the paper discusses the two methods for characterizing safe controllers which were outlined above. The first method will be referred to as the LP-method since it solves a sequence of linear programs to determine safe control signals. The LP-method is discussed in section 3. The fundamental result in section 3 is a set of inequality constraints characterizing locally safe piecewise constant control signals. The second method is referred to as the MRC-method since it uses a model reference control (MRC) approach to formulate the controller synthesis problem. The MRC method is discussed in section 4. The fundamental results in this section are sufficient conditions characterizing controllers ensuring bounded-amplitude performance for the switched control system. Section 5 compares both methods and draws some general conclusions about their relative strengths and weaknesses.

2 Safe Supervisory Controllers

Hybrid dynamical systems arise when the time and/or the state space have mixed continuous and discrete natures. Such systems frequently arise when computers are used to control continuous state systems. In recent years, specific attention has been focused on hybrid systems in which a discrete-event system is used to supervise the behavior of plants whose state spaces are dense in $\Re^n$. In this class of hybrid control systems, commands are issued by a discrete-event system to direct the behavior of the plant. These commands are high-level directives to the plant which require that the supervised plant satisfy logical conditions on the plant's state. The simplest set of conditions are inequality conditions on the plant's state.

Assume that the plant's dynamics are generated by the differential equation

$$\dot{x} = f(x, u) \tag{1}$$

where $x \in \Re^n$ is the state, $u \in \Re^m$ is the control input, and $f : \Re^n \times \Re^m \to \Re^n$ is a Lipschitz continuous mapping. A *supervisory directive* to this system is characterized by a set of functionals, $h_j : \Re^n \to \Re$ for $j = 0, \ldots, N$, that separate the state space. The functionals, h_j, are said to *separate* the state space if and only if for all $x, y \in \Re^n$ such that $h_j(x) > 0$ and $h_j(y) < 0$, there exists $0 < \lambda < 1$ such that $h_j(\lambda x + (1-\lambda)y) = 0$. The functional, h_0, is said to be the *goal trigger* and the other functionals, h_j for $j = 1, \ldots, N$, are called the *guard triggers*. Consider a state feedback controller,

$$u = k(x) \tag{2}$$

Such a controller is said to be *safe* if and only if there exist finite times T_1 and T_2 ($T_1 < T_2$) such that

- $h_j(x(t)) < 0$ for all $t_0 \leq t < T_2$ ($j = 1, \ldots, N$),
- $h_0(x(t)) < 0$ for all $t_0 \leq t < T_1$,
- and $h_0(x(t)) > 0$ for all $T_1 < t < T_2$.

Essentially, these conditions state that the goal condition is triggered in finite time without any of the guard triggers being violated. Assume that we have a monotone increasing function $r(t)$ such that $r(0) = h_0(x(0))$ and $r(T_1) = 0$. We can use this "reference" function to rewrite the preceding list of conditions as a set of inequality constraints such that the guard triggers ($j = 1, \ldots, N$) satisfy $h_j(x(t)) < 0$ and the goal trigger satisfies $h_0(x(t)) - r(t) > 0$ for all $t \in [0, T_2]$. However, note that with this setting, the switching time for h_0 is less than T_1.

3 LP-Method

The LP-method is motivated by a hybrid system architecture outlined in [Kohn 1993]. This method characterizes safe control signals as a set of linear inequality constraints. The LP-method assumes that the plant's differential equation has the form

$$\dot{x} = f_0(x) + \sum_{i=1}^{m} f_i(x)u_i(t) \tag{3}$$

where $f_i : \Re^n \rightarrow \Re^n$ are analytic functions forming a nonsingular distribution of vector fields in $\Re^n$. It is also assumed that the set of trigger functions $\{h_j\}_{j=1}^{N}$ is analytic.

Assume that the trigger functions, $h_j(x(t))$, are known at time, t. Under appropriate conditions, it is possible to represent the trigger functions at time $t + \delta$ as a Fliess functional series. To formally state these results, some notational conventions need to be introduced. Let $f : \Re^n \rightarrow \Re^n$ be a vector of analytic functions, $f' = \left[\begin{array}{cccc} f_1 & f_2 & \cdots & f_n \end{array} \right]$ where $f_i : \Re^n \rightarrow \Re^n$ $(i = 1, \ldots, n)$. The Lie derivative of an analytic function $h : \Re^n \rightarrow \Re$ with respect to vector field f is

$$L_f h(x) = \sum_{i=1}^{n} \frac{\partial h}{\partial x_i} f_i(x) \tag{4}$$

Let $i \in \{1, \ldots, m\}$ be an *index* and let $i_1, \ldots, i_k$ be a sequence of indices of length k called a *multi-index*. The set of all multi-indices will be denoted as I^*. Associated with the multi-index $i_1, \ldots, i_k$ is the iterated integral,

$$E_{i_k, \ldots, i_1}(t) = \int_0^t d\xi_{i_k} \ldots \xi_{i_1} \tag{5}$$

where for $i = 1, \ldots, m$,

$$\xi_i(t) = \int_0^t u_i(\tau) d\tau \tag{6}$$

$$\int_0^t d\xi_{i_k} \ldots d\xi_{i_1} = \int_0^t d\xi_{i_k}(\tau) \int_0^\tau d\xi_{i_{k-1}} \ldots d\xi_{i_1} \tag{7}$$

The following theorem which is proven in [Isidori 1989] will be used in our following development.

Proposition 1 *[Isidori 1989](pp 114-119) Consider the system given by equations 3. If there exist $K > 0$ and $M > 0$ such that*

$$\left|L_{f_{i_1}} \ldots L_{f_{i_k}} h_j(x(t))\right| \leq K k! M^k \tag{8}$$

for all k, j, and all multi-indices in I^*, then there exists a real $\Delta > 0$ such that for all $\delta \in [0, \Delta]$ and piecewise continuous control functions $u_i(t)$ defined over $[t, t+\Delta]$ subject to the constraint

$$\max_{\delta \in [0,\Delta]} |u_i(t+\delta)| < 1 \tag{9}$$

then the series

$$h_j(x(t)) + \sum_{k=1}^{\infty} \sum_{I^*} L_{f_{i_1}} \ldots L_{f_{i_k}} h_j(x(t)) \int_0^{\delta} d\xi_{i_k} \ldots d\xi_{i_1} \tag{10}$$

is uniformly and absolutely convergent to $h_j(x(t+\delta))$.

If we can find a control signal, u, so that the safety conditions are satisfied over $[t, t+\delta]$, for all t then we say the control is locally safe. The Fliess series is a formal series over the control symbols, u_i. It provides a means of expressing the values of the trigger functions, h_j, over a finite interval, $[t, t+\delta]$. It therefore makes sense to use the Fliess series in characterizing control inputs, u_i, ensuring local safety of the control system. The following proposition provides just such a characterization.

Proposition 2 *Consider the system given by equation 3 and let $r(t)$ be a known reference trigger such that $\dot{r}(t) = R > 0$ and $r(0) = h_0(x(0))$. Assume that proposition 1 holds and that $x(0)$ is safe. If there exist $\gamma > 0$, $\gamma_1 > 0$, and $\Delta > 0$ such that the constant vector $u^* \in \Re^m$ satisfies*

$$-\gamma > h_j(x(0)) + \sum_{i=1}^{m} [L_{f_i} h_j(x(0))]\, u_i^* \Delta \quad (j = 1, \ldots, N) \tag{11}$$

and

$$|u_i^*| \leq 1 \quad i = 1, \ldots, m \tag{12}$$

and

$$R - \gamma_1 < \sum_{i=1}^{m} [L_{f_i} h_0(x(0))]\, u_i^* \tag{13}$$

then the constant control $u(t) = u^*$ *generates a safe state trajectory in* $[0, \Delta)$.

Proof: Assuming that proposition 1 holds,then there exists $K > 0$ and $M > 0$ such that the growth constraint 8 is satisfied. Given inequality 12, we know that the Fliess series is uniformly convergent in an interval, $[0, \Delta]$, and that for any $\delta \in [0, \Delta]$, we can expand $h_j(x(\delta))$ as

$$h_j(x(\delta)) = h_j(x(0)) + \sum_{k=1}^{\infty} \sum_{i_1,\ldots,i_k} L_{f_{i_1}} \cdots L_{f_{i_k}} h_j(t) E_{i_k,\ldots,i_1}(\delta) \tag{14}$$

Assuming a constant u^* over this interval, we see that

$$\begin{aligned} h_j(x(\delta)) &= h_j(x(0)) + \sum_{k=1}^{\infty} \sum_{i_1,\ldots,i_k} L_{f_{i_1}} \cdots L_{f_{i_k}} h_j(x(0)) u_{i_1}^* \cdots u_{i_k}^* \frac{\delta^k}{k!} \\ &= h_j(x(0)) + \sum_{i=1}^{m} L_{f_i} h_j(x(0)) u_i^* \delta + o_j(\delta) \end{aligned} \tag{15}$$

The tail term is

$$o_j(\delta) = \sum_{k=2}^{\infty} \sum_{i_1,\ldots,i_k} L_{f_{i_1}} \cdots L_{f_{i_k}} h_j(x(0)) u_{i_1}^* \cdots u_{i_k}^* \frac{\delta^k}{k!} \tag{16}$$

The magnitude of the tail is bounded as

$$|o_j(\delta)| \leq K(Mm\delta)^2 \left(\frac{1}{1 - Mm\delta}\right) \tag{17}$$

for $\delta < 1/Mm$.

We now take $\Delta = \rho/Mm$ where $\rho < 1$, then

$$|o_j(\delta)| \leq K \frac{\rho^2}{1 - \rho} = \gamma \tag{18}$$

We take the right-hand side of this inequality to be the γ of our theorem and immediately conclude that inequality 15 can be written as

$$h_j(x(\delta)) \leq h_j(x(0)) + \sum_{i=1}^{m} L_{f_i} h_j(x(0)) u_i^* \Delta + \gamma \tag{19}$$

For $j = 1, \ldots, N$, this implies that the state is safe at time Δ. It is also safe at time zero. Since our bound is linear this must also hold for all δ between 0 and Δ. So for all time in $[0, \Delta)$, the desired inequality constraints ensure the guard triggers are not violated.

We now turn to the terminating trigger, $h_0(x(t))$. In this case, we require that $h_0(x(\delta)) > r(\delta)$ for all $\delta \in [0, \Delta]$. By assumption, $h_0(x(0)) \geq r(0)$ and we know by that $r(\delta) = r(0) + R\delta$. To ensure our other constraint is satisfied, we require

$$r(0) + R\delta < h_0(x(0)) + \sum_{i=1}^{m} L_{f_i} h_0(x(0)) u_i^* \delta + K(Mm\delta)^2 \left(\frac{1}{1 - Mm\delta}\right) \tag{20}$$

Assuming that $r(0) = h_0(x(0))$, we see that the condition reduces to

$$R < \sum_{i=1}^{m} L_{f_i} h_0(x(0)) u_i^* + KMm\frac{\rho}{1-\rho} \tag{21}$$

We treat this last quantity as γ_1, and our result follows. **QED**

Proposition 2 characterizes the *set* of locally safe control signals. In practice, a specific control signal will need to be chosen from this set. This selection is made with respect to an assumed cost functional, $J(u)$. The "optimal" locally safe control is determined by finding the control signal that minimizes this given cost subject to the local safety conditions represented by the inequality constraints in proposition 2. A particularly simple choice for the cost is a linear function of u. If we restrict $0 < u_i < 1$ for all $i = 1, \ldots, m$, then our cost functional becomes

$$J(u) = w'u = \sum_{i=1}^{m} w_i u_i \tag{22}$$

where w is an m-vector of positive weights. The control signal minimizing this cost is obtained by solving the following linear programming problem

$$\begin{array}{ll} \text{minimize:} & w'u \\ \text{with respect to:} & u \\ \text{subject to:} & A(t)u < b \\ & 0 < u_i < 1 \end{array} \tag{23}$$

where

$$A(t) = \begin{bmatrix} -L_{f_1}h_0 & -L_{f_2}h_0 & \cdots & -L_{f_m}h_0 \\ L_{f_1}h_1 & L_{f_2}h_1 & \cdots & L_{f_m}h_1 \\ \cdot & \cdot & \cdots & \cdot \\ L_{f_1}h_n & L_{f_2}h_n & \cdots & L_{f_m}h_n \end{bmatrix} \tag{24}$$

and

$$b = \begin{bmatrix} -R + \gamma_1 \\ -\gamma - h_1(x(0)) \\ \cdot \\ -\gamma - h_n(x(0)) \end{bmatrix} \tag{25}$$

Note that the constraint matrix $A(t)$ is a function of time.

The preceding discussion solved an LP problem to find a constant control, u^*, for a time $t \in [0, T)$ which was locally safe. A safe control trajectory, $u^*(t)$, for all $t \in [0, T)$, can be determined by solving a sequence of linear programs at the time instants $t_0 + n\Delta$, where n is the set of positive integers and Δ is given by the growth constants in proposition 2. The constraint matrices $A(t)$ are obtained from our knowledge of the distribution, $\{f_0, f_1, \ldots, f_n\}$ as well as the current state vector. This essentially means that an LP problem must be solved at the sampling instant $t_0 + n\Delta$ to determine the piecewise constant control u^* that is used over the interval $[t_0 + n\Delta, t_0 + (n+1)\Delta]$.

The solution u^* will ensure the safety of the trajectory over the interval $[t_0 + n\Delta, t + 0 + (n+1)\Delta]$. Will the concatenation of these u^* yield a safe system? The answer is "yes" provided $A(t)$ does not change "too quickly" over the generated state-space trajectory. Recall from the proof of proposition 2 that $\Delta < 1/Mm$ where m is the number of applied inputs and M is the bounding constant given in the growth condition of equation 8. Assume that the growth condition is uniformly satisfied for all points along the state trajectory, then there exists a single M bounding all Lie derivatives in equation 8 and we

see that Δ is fixed. In this case we can clearly ensure the safety of the concatenated set of controls.

Example: A simple example is used to illustrate the approach. The following example has been modified from [Deshpande 1995] to yield a plant which is affine in the control. The modified plant equations are

$$\dot{x}_1 = -x_1 + (u_1 - u_2) \tag{26}$$

$$\dot{x}_2 = -x_2 + (1 + x_1^2)(u_3 - u_4) \tag{27}$$

where u_i is constrained to be non-negative for $i = 1, \ldots, 4$. This vector field clearly satisfies the growth conditions of equation 8, so we can apply our method to safely control this system. We can rewrite this as a linear combination of vector fields,

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} -x_1 \\ -x_2 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u_1 + \begin{bmatrix} -1 \\ 0 \end{bmatrix} u_2 + \begin{bmatrix} 0 \\ 1 + x_1^2 \end{bmatrix} u_3 + \begin{bmatrix} 0 \\ -(1 + x_2)^2 \end{bmatrix} u_4 \tag{28}$$

The control objective is to move the plant from an initial state near the operating point $(0, 0)$ to a point near $(2.5, 2)$. Note that the control inputs have been paired so there is a "positive" input (u_1 and u_3) and a "negative" input (u_2 and u_4) which work in opposition to each other.

The guard triggers are

$$h_1(x) = x_2 - 1.25x_1 - .5 \tag{29}$$

$$h_2(x) = x_2 - 1.25x_1 + .5 \tag{30}$$

with a goal trigger,

$$h_0(x, t) = x_1(0) - Rt \tag{31}$$

where R is the desired rate at which we want to achieve the desired goal set. In this example $R = 0.1$. These regions are illustrated in figure 1.

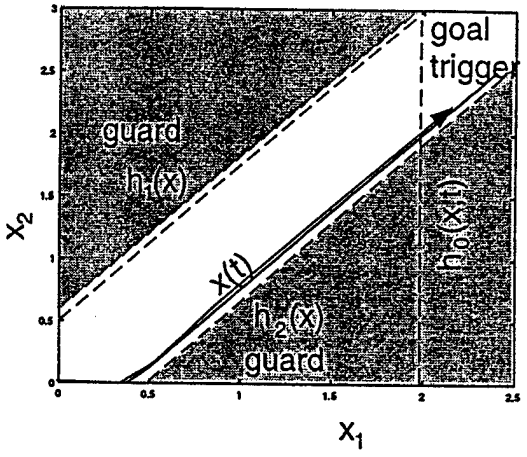


Figure 1: Guard and goal triggers for example.

A simple Matlab script was written to simulate this system. Figure 1 illustrates the state trajectory that was generated by this approach. In this case, the LP-problems determining safe controls were computed at a rate $\Delta = 0.1$. The weighting vector w was chosen to be a vector of ones. As can be seen, the selected controls basically select one control strategy that drives the system in the direction of the h_2 guard trigger. Once within a distance γ of that guard trigger, the control strategy changes to a chattering policy which drives the system state along the boundary until the terminal condition is satisfied. The chattering nature of the control policy is seen in figure 2.

This example illustrates some fundamental characteristics of the LP-approach to safe controller generation. In the first place, this is an on-line procedure which requires the solution of an LP problem at each sampling instant. The computation of the control requires significant information about the underlying vector fields generating the system's dynamics. Finally, this approach tends to produce a chattering control strategy, as shown in figure 2.

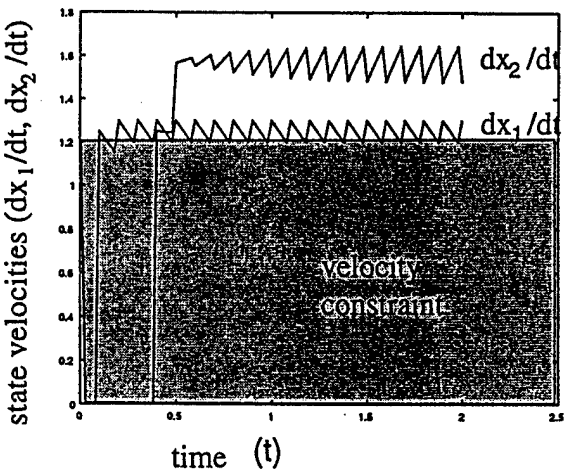


Figure 2: Chattering control policy

4 MRC-Method

A model reference control (MRC) approach for implementing safe controllers was introduced in [Lemmon 1996]. In this approach, the plant is forced to follow a reference trajectory, $x_m(t)$, which is known to be safe with a worst case tracking error of γ. Provided there exists a time T such that $h_0(x_m(T)) > \gamma$ and for all $0 < t < T$ and $j = 1, \ldots, N$ that $h_j(x_m(t)) > -\gamma$, the plant trajectory, $x_p(t)$,is guaranteed to be safe.

In this framework, synthesis of safe switched controllers is accomplished by examining the error between the plant and reference trajectories. Suppose that the plant state dynamics are generated by

$$\dot{x}_p = f_p(x_p, u) \tag{32}$$

and let the reference trajectory be generated by

$$\dot{x}_m = f_m(x_m) \tag{33}$$

Defining the state error signal, $x = x_m - x_p$, yields the differential equation

$$\dot{x} = f(x_m, x, u) = f_m(x_m) - f_p(x_p, u) \tag{34}$$

The control input is generated by a controller $u = k(x_m, x)$ which is dependent on the reference model state and the reference error.

One control strategy is to choose a collection of setpoints along the reference trajectory, $x_m(t)$, and design linear control agents at each of the setpoints using the plant model obtained from linearizing about the corresponding setpoint. This is the basic idea behind the switched linear control agent approach introduced in [Lemmon 1996]. Note that, as in a classical gain scheduling approach (see, e.g. [Shamma 1990]), each of the control agents designed using this approach is designed for local performance near an associated setpoint. As with classical gain scheduling, performance of the switched system will be difficult to guarantee, in general, due to the approximations made in the setpoint linearizations (as well as other modeling uncertainties). Thus, the linear setpoint controllers should, at the least, demonstrate robustness to the system nonlinearities lost in the setpoint linearizations. One way of incorporating this robustness requirement into the design is to use linear parameter varying (LPV) plant models at each of the setpoints.

An LPV model of the error dynamics may be obtained by rewriting the dynamics of equation 34 as

$$\dot{x} = A(\theta)x + B_u(\theta)u + B_w(\theta)w \tag{35}$$

$$z = Cx + Du \tag{36}$$

where $w = 1$ is introduced as a fictitious disturbance. The s-dimensional parameter vector, θ, is a function $S(x_m, x, u)$. The vector θ is assumed to vary continuously over a compact subset $\tilde{\Theta} \subset \Re^s$; this assumption is denoted $\theta \in \mathcal{F}_{\tilde{\Theta}}$. For each of the local plant models, θ is assumed to vary continuously over a compact subset $\Theta \subseteq \tilde{\Theta}$ for a time interval $[\tau_s, \tau_f]$; this assumption is denoted $\theta \in \mathcal{F}_{\Theta}[\tau_s, \tau_f]$. This notation distinguishes a *parameter variation* over Θ from a point in Θ which will be denoted $\theta \in \Theta$. The vector z will be called the *objective* signal and is chosen (via C and D) to reflect not only the

the trigger constraints, but also control energy constraints. The entire LPV system will be denoted as $\Sigma(\tilde{\Theta}, A, B, C, D)$ where $B' = [B'_u \; B'_w]$.

Let $\mathcal{T}_i = [t_i, t_{i+1})$ denote the time interval over which the ith setpoint controller is used. Note that if each individual setpoint controller satisfies the performance requirement,

$$\sup_{t \in \mathcal{T}_i} \|z(t)\| < \gamma \tag{37}$$

then local safety of the control directive will be preserved. Local setpoint controllers are therefore obtained by solving what is called a finite horizon $\mathcal{L}_1$, or *bounded-amplitude*, optimal control problem for LPV systems.

There are, unfortunately, relatively few results for the solution of $\mathcal{L}_1$ optimal control problems. In [Dahleh 1987], it was shown that optimal solutions to this problem are irrational or infinite dimensional, even for rational and finite-dimensional plants. For deterministic linear time-invariant systems [Nagpal 1994] an approach to $\mathcal{L}_1$ optimal control synthesized a sub-optimal controller minimizing an upper bound on the bounded-amplitude gain by solving a set of linear matrix inequalities. To use this prior work in our synthesis problems, however, existing synthesis methods must be extended to LPV systems. We remark here that previous results on gain scheduling for LPV systems (e.g. [Shamma 1991]) do not directly apply to the performance problem introduced here because those results apply to an $\mathcal{L}_2$ performance measure. The following theorem provides a characterization of systems whose $\mathcal{L}_1$ gains are bounded.

Note: The remainder of the paper will use the following notation: the infinite-horizon ∞-norm of a signal $x(t)$ is defined as $\|x(t)\|_\infty := \sup_t \|x(t)\|$ where $\|\cdot\|$ is the Euclidean norm. $\mathcal{L}_\infty^n$ is the space of n-dimensional vector signals with finite ∞-norm; $\mathcal{BL}_\infty^n$ is the space of n-dimensional vector signals with ∞-norm bounded by 1. For constants $\tau < T$, finite-horizon ∞-norm of a signal $x(t)$ defined on the interval $[\tau, T]$ is $\|x(t)\|_{\infty,[\tau,T]} := \sup_{t \in [\tau,T]} \|x(t)\|$. $\mathcal{L}_\infty^n[\tau, T]$ and $\mathcal{BL}_\infty^n[\tau, T]$ are defined in an analogous

manner. Recall that $\theta \in \mathcal{F}_\Theta[\tau, T]$ is an s-dimensional signal $\theta(t)$ which takes values on a compact subset $\Theta \subset \Re^s$ for $t \in [\tau, T]$. This implies that $\theta \in \mathcal{L}_\infty^s[\tau, T]$ Finally, throughout the remainder of the paper, the matrix inequality $M > N$ ($M \geq N$) where M and N are symmetric matrices, indicates that the matrix $M - N$ is positive definite (positive semi-definite).

Proposition 3

Given constants $r > 0$, $\gamma > 0$ and $T > 0$ and the LPV system $\Sigma(\tilde{\Theta}, A, B, C, D)$ with $u = 0$. Let Θ be a compact subset of $\tilde{\Theta}$ and suppose there exists $\alpha > 0$ and $\beta \geq 0$ and a positive definite matrix $P \in \Re^{n \times n}$ satisfying

$$P \geq \frac{r}{\gamma^2} C'C \tag{38}$$

and

$$A'(\theta)P + PA(\theta) + \left(2\beta + \frac{\alpha}{r}\right) P + \frac{1}{\alpha} PB_w(\theta)B_w(\theta)'P \leq 0 \tag{39}$$

for all $\theta \in \Theta$. If $\theta \in \mathcal{F}_\Theta[0, T]$ and $w \in \mathcal{BL}_\infty^{n_w}[0, T]$, then

- *if $x'(0)Px(0) \leq r$ then $x'(t)Px(t) \leq r$ and $z'(t)z(t) \leq \gamma^2$ for all $t \in [0, T]$*
- *if $\beta > 0$ and $x'(0)Px(0) = r_0 > r$, then $x'(t)Px(t) \leq r$ for all $t \in [t_d, T]$ where*

$$t_d := -\frac{1}{2\beta} \log\left(\frac{r}{r_0}\right) \tag{40}$$

(assuming $t_d \leq T$).

Proof: Let $r > 0$, $\gamma > 0$ and $T > 0$ and assume there are constants $\alpha > 0$ and $\beta \geq 0$ and a positive definite matrix P so that the conditions of the theorem are satisfied. For any $\theta \in \Theta$,

$$\frac{1}{\alpha} PB(\theta)B'(\theta)P \geq 0 \tag{41}$$

If equation 39 holds for all $\theta \in \Theta$, then

$$A'(\theta)P + PA(\theta) + \left(\frac{\alpha}{r} + 2\beta\right) P \leq -\frac{1}{\alpha} PB(\theta)B'(\theta)P \leq 0 \tag{42}$$

Using Schur complements, this inequality is true if and only if

$$\begin{bmatrix} A'(\theta)P + PA(\theta) + \rho P & PB(\theta) \\ B'(\theta)P & -\alpha I \end{bmatrix} \leq 0 \tag{43}$$

where $\rho = 2\beta + \alpha/r$. This inequality is true if and only if

$$\begin{bmatrix} \xi \\ \nu \end{bmatrix}' \begin{bmatrix} A'(\theta)P + PA(\theta) + \rho P & PB(\theta) \\ B'(\theta)P & -\alpha I \end{bmatrix} \begin{bmatrix} \xi \\ \nu \end{bmatrix} \leq 0 \tag{44}$$

for all $\xi \in \Re^n$ and $\nu \in \Re^{n_w}$. Expanding, it is apparent that

$$\xi' \left[A'(\theta)P + PA(\theta) + 2\beta P\right] \xi + \nu' B'(\theta) P\xi + \xi' PB(\theta)\nu + \frac{\alpha}{r} \left[\xi' P\xi - r\right] + \alpha \left[1 - \nu'\nu\right] \leq 0 \tag{45}$$

This last equation implies that

$$\xi' \left[A'(\theta)P + PA(\theta)\right] \xi + \nu' B'(\theta) P\xi + \xi' PB(\theta)\nu \leq -2\beta \xi' P\xi \leq 0 \tag{46}$$

for all ξ and ν such that $\xi' P\xi \geq r$ and $\nu'\nu \leq 1$.

Now consider a function, $V : \Re^n \to \Re$, such that $V(\xi) = \xi' P\xi$. Along trajectories of the LPV system with $u = 0$, the time derivative of $V(x(t))$ is

$$\frac{dV}{dt}(x(t)) = x'(t) \left[A'(\theta(t))P + PA(\theta(t))\right] x(t) + w'(t)B'(\theta(t))Px(t) + x'(t)PB(\theta(t))w(t) \tag{47}$$

and from equation 46, it is immediately evident that

$$\frac{dV}{dt}(x(t)) \leq -2\beta V(x(t)) \leq 0 \tag{48}$$

for any $x(t)$ and $w(t)$ such that $x'(t)Px(t) \geq r$ and $w'(t)w(t) \leq 1$ with $t \in [0, T]$.

Assume, for some $w \in \mathcal{BL}_\infty^{n_w}[0, T]$, that there is a trajectory with initial state $x(0)$ satisfying $V(x(0)) = x'(0)Px(0) \leq r$ and $V(x(T)) > r$. Since $V(x(t))$ is differentiable in

t, the mean value theorem may be used to imply the existence of a time $\tau \in [0,T]$ such that $V(x(\tau)) \geq r$ and $\dot{V}(x(\tau)) > 0$. This is a contradiction of equation 48, so one must conclude that $x'(t)Px(t) \leq r$, hence $z'(t)z(t) \leq \gamma^2$, for all $t \in [0,T]$.

If $V(x(0)) > r$, then the differential inequality implies that

$$V(x(t) \leq V(x(0)) - \int_0^t 2\beta V(x(\tau))d\tau \tag{49}$$

and the Bellman-Gronwall inequality may be used to conclude that

$$V(x(t)) \leq V(x(0))e^{-2\beta t} \tag{50}$$

Now suppose that $V(x(0)) = r_0 > r$, $\beta > 0$ and let t_d be the *dwell time* given in equation 40. If $t_d \leq T$, then

$$V(x(t)) \leq r_0 e^{-2\beta t_d} = r \tag{51}$$

for all $t \in [t_d, T]$. •

Proposition 3 characterizes a class of uncontrolled ($u = 0$) LPV systems where $\|z\|_{\infty,[0,T]} \leq \gamma$ and where the parameter variation is confined to the set Θ. The next result helps characterize a class of controlled LPV systems using linear state feedback, $u = Kx$.

Proposition 4

Given $\gamma > 0$ and an LPV system $\Sigma(\tilde{\Theta}, A, B, C, D)$ with state space realization

$$\begin{bmatrix} \dot{x}(t) \\ z(t) \end{bmatrix} = \left[\begin{array}{c|cc} A(\theta) & B_w(\theta) & B_u(\theta) \\ \hline C & 0 & D \end{array}\right] \begin{bmatrix} x \\ w \\ u \end{bmatrix} \tag{52}$$

Let $\Theta \subseteq \tilde{\Theta}$ be a compact subset and consider a state feedback control law $u = Kx$ where $K \in \Re^{n_u \times n}$. Define $\tilde{A}(\theta) = A(\theta) + B_u(\theta)K$. Then, there exist constants $\alpha_1 \geq \alpha_2 > 0$, a positive definite matrix $P \in \Re^{n \times n}$ and a controller K satisfying

$$P \geq \frac{1}{\gamma^2}(C + DK)'(C + DK) \tag{53}$$

and

$$\tilde{A}'(\theta)P + P\tilde{A}(\theta) + \alpha_1 P + \frac{1}{\alpha_2} PB_w(\theta)B_w(\theta)'P \leq 0 \tag{54}$$

for all $\theta \in \Theta$ if and only if there exists a positive definite matrix $Q \in \Re^{n\times n}$ and a matrix $V \in \Re^{n_u \times n}$ such that for all $\theta \in \Theta$

$$\begin{bmatrix} Q & QC' + V'D' \\ CQ + DV & \gamma^2 I \end{bmatrix} \geq 0 \tag{55}$$

and

$$QA'(\theta) + A(\theta)Q + \alpha_1 Q + \frac{1}{\alpha_2} B_w(\theta)B_w'(\theta) + B_u(\theta)V + V'B_u'(\theta) \leq 0 \tag{56}$$

Proof: Assume that there exists a positive definite matrix Q and a matrix V such that

$$\begin{bmatrix} Q & QC' + V'D' \\ CQ + DV & \gamma^2 I \end{bmatrix} \geq 0 \tag{57}$$

Using Schur complements, this holds if and only if

$$Q - \frac{1}{\gamma^2}(QC' + V'D')(CQ + DV) \geq 0 \tag{58}$$

If we let $P = Q^{-1}$ and $K = VQ^{-1}$, then this holds if and only if

$$P \geq \frac{1}{\gamma^2}(C + DK)'(C + DK) \tag{59}$$

which establishes the first condition in the proposition.

Now assume that there also exist constants $\alpha_1 \geq \alpha_2 > 0$ such that

$$\tilde{A}'(\theta)P + P\tilde{A}(\theta) + \alpha_1 P + \frac{1}{\alpha_2} PB_w(\theta)B_w(\theta)'P \leq 0$$

for all $\theta \in \Theta$. Substituting $P = Q^{-1}$ and $K = VQ^{-1}$ as above,

$$[A(\theta) + B_u(\theta)K]'P + P[A(\theta) + B_u(\theta)K] + \alpha_1 P + \frac{1}{\alpha_2} PB_w(\theta)B_w'(\theta)P \tag{60}$$

$$= Q^{-1}\left[QA'(\theta) + A(\theta)Q + \alpha_1 Q + \frac{1}{\alpha_2} B_w(\theta)B_w'(\theta) + B_u(\theta)V + V'B_u'(\theta)\right]Q^{-1} \tag{61}$$

Since $Q^{-1} > 0$, the conclusion of the theorem immediately follows. •

The following remarks summarize the importance of propositions 3 and 4.

- Under the assumptions of proposition 3 and 4, it should be apparent that if the inequalities 55 and 56 hold, then under control $u = Kx$, the objective function $z = (C + DK)x$ will have a finite horizon sup-norm less than γ provided that the parameter variation is bounded according to $\theta \in \mathcal{F}_\Theta[0, T]$.

- From the proof of proposition 4 it should be apparent that the matrices Q and V satisfying inequalities 55 and 56 parameterize a set of locally safe controllers. In particular, for any such Q and V, the controller is $K = VQ^{-1}$.

- The importance of inequalities 55 and 56 is that these can be used to form matrix inequalities which are *linear* in Q and V. These inequalities need only be satisfied *pointwise* over Θ without regard to parameter variation rate, as long as the parameter variation is bounded according to $\theta \in \mathcal{F}_\Theta[0, T]$.

- Note that the θ dependence of inequality 56 limits its usefulness: verifying the condition for all $\theta \in \Theta$ may be unreasonable or infeasible. In certain cases, however, the computational burden can be significantly reduced. For instance, if $A(\theta)$, $B_u(\theta)$ and $B_w(\theta)$ can be written as linear fractional transformations in θ, and if the parameter set Θ is a polytope, then it is possible to express inequality 56 as a matrix inequality which is independent of θ and linear in the variables Q and V. Derivation of such LMIs is a straightforward application of the results in [Boyd 1994]; a detailed proof is beyond the scope of this paper but can be found in [Bett 1997].

The results in proposition 3 are extremely important in determining whether or not a given set of linear setpoint controllers will safely execute a supervisory directive. Let $\mathcal{T}_i$ be the time interval when the ith setpoint controller is used. This controller is characterized by the matrices P_i, the radius r_i, and constants, α_i and β_i. The results in this proposition state that the controlled system will be locally safe if the error satisfies $x'(t_i)P_i x(t_i) \leq r_i^2$.

To ensure that the plant behavior is safe under the next ($i+1$st) setpoint controller, one must ensure that $x'(t_{i+1})P_{i+1}x(t_{i+1}) \leq r_{i+1}^2$. The problem here is that the second condition is not guaranteed if the switch occurs too quickly. This is where the second part of proposition 3 has something to add. Specifically, if the state at time t_i starts outside of the invariant set for the $i+1$st setpoint controller, then there is a minimum time called the dwell time, after which the state is guaranteed of being within the required distance. In particular, let $r_i P_{i+1} \leq r_{i+1} P_i$ and assume that P_i and P_{i+1} both satisfy the conditions for setpoint controllers in proposition 3. It is readily apparent that if $t_{i+1} - t_i \geq t_d$, where

$$t_d = -\frac{1}{2\beta_i} \log \frac{r_{i+1}}{r_i}, \tag{62}$$

then

$$\|z\|_{\infty,[t_i,t_{i+2}]} \leq \gamma \tag{63}$$

The satisfaction of the inequality constraints, of course, also requires that $\theta(t)$ lie in Θ_1 for $t_i \leq t < t_{i+1}$ (i.e. $\theta \in \mathcal{F}_{\Theta_1}[t_i, t_{i+1}]$) and in Θ_2 for $t_{i+1} \leq t \leq t_{i+2}$. Satisfaction of this parameter variation condition is non-trivial to verify.

The preceding discussion has outlined how the conditions determined in proposition 3 can be used to ensure safe behavior between the switch of two different setpoint controllers. These conditions are summarized in the following proposition.

Proposition 5 (LPV Switching Lemma)

Given LPV systems $\Sigma(\tilde{\Theta}, A_1, B_1, C_1, D_1)$, and $\Sigma(\tilde{\Theta}, A_2, B_2, C_2, D_2)$ with associated controllers K_1 and K_2, let the ith controller ($i = 1, 2$) be characterized by matrix P_i, and positive constants, r_i, α_i, and β_i so that the conditions of propositions 3 and 4 are satisfied for compact parameter sets $\Theta_i \subseteq \tilde{\Theta}$. Assume that controller K_1 is used over time interval $t \in [t_0, t_s)$ and that controller K_2 is used for time interval $t \in [t_s, T]$ for any

$T > t_s$. *If* $r_1 P_2 \leq r_2 P_1$ *and the switch time* t_s *satisfies*

$$t_s - t_0 \geq -\frac{1}{2\beta_1} \log \frac{r_2}{r_1}, \tag{64}$$

and if $\theta \in \{\mathcal{F}_{\Theta_1}[t_0, t_s], \mathcal{F}_{\Theta_2}[t_s, T]\}$ *then* $\|z\|_{\infty,[0,T]} \leq \gamma$.

The LPV switching lemma suggests a means of testing to see whether or not a given collection of linear setpoint controllers will generate a safe trajectory. Essentially, this involves verifying the dwell-time condition for all possible switching times and verifying the conditions on the parameter variation. The required dwell-times may be computed from the synthesis LMIs and the coupling condition $r_1 P_2 \leq r_2 P_1$. Switching times and parameter variation bounds are more difficult to verify, but a *nominal parameter trajectory*, $S(x_m(t), 0, 0)$, may be used to estimate these quantities off-line. These estimates may then be compared to the dwell-time results as a sufficient condition for safeness.

Example: As an illustration of some of the important aspects of the MRC-approach, the methods described above were applied to the process control example described in section 3. The reference model

$$\begin{aligned} \dot{x}_{m1} &= x_{m1} - 1.63 - x_{m1}/(1 + 0.5 \sin 10(x_{m1} - 1.63)) \\ \dot{x}_{m2} &= 1 \end{aligned} \tag{65}$$

is specified to move the plant from an initial state near the operating point $(x_{p1}, x_{p2}) = (2.5, 2)$ to a point near $(x_{p1}, x_{p2}) = (1, 3)$ in 1 second. The performance weights for the objective function were chosen as $C = I$ and $D = 0.1I$; the desired bound on the objective function was $\gamma = 0.5$.

The LPV error system is derived as

$$\begin{aligned} \dot{x} &= A(\theta)x + B_u(\theta)u + B_w(\theta)w \\ z &= Cx + Du \end{aligned} \tag{66}$$

with

$$A(\theta) = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix}, \; B_w(\theta) = \begin{bmatrix} \theta_1 - 1.63 \\ \theta_2 + 1 \end{bmatrix} \text{ and } B_u(\theta) = \begin{bmatrix} -1 & 0 \\ 0 & -\theta_3 - 1 \end{bmatrix}, \quad (67)$$

and parameter mapping

$$\begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \end{bmatrix} = S(x_m, x, u) = \begin{bmatrix} 2x_{m1} - x_{m1}/(1 + 0.5 \sin 10(x_{m1} - 1.63)) \\ x_{m2} + (2x_{m1}x_1 - x_1^2)u_2 \\ x_{m1}^2 \end{bmatrix}. \quad (68)$$

Linear state feedback control agents, $u = Kx$, were designed by choosing setpoints, $\theta^{nom'} = S(x_m^{nom}, 0, 0)$ and solving the appropriate synthesis LMIs for parameter sets

$$\Theta := \left\{ \theta \left| \sup_{i=1,\ldots,s} |\theta_i - \theta_i^{nom}| \leq \vartheta \right. \right\}$$

for a design parameter $\vartheta > 0$. Switching control was achieved by switching a new feedback controller into the loop whenever the parameter variation evolved onto the boundary of the current agent's parameter set. The new control agent was chosen to minimize a distance measure in the parameter space.

A Matlab program was written to solve the appropriate synthesis LMIs, as indicated above, and simulate the closed loop system. Simulations were performed for various values of ϑ, resulting in experiments requiring varying numbers of models. The resulting state trajectory for twenty models is depicted in figure 3. Figure 3 also depicts the reference trajectory and forbidden (shaded) regions of the state space. Note that the resulting trajectory is safe and non-chattering, as seen in figure 4. Similar results were observed for different numbers of models. Figure 5 depicts the resulting trend observed for increasing numbers of models. Note the monotonic improvement in performance with increasing numbers of models. The quantity $\|\theta_{err}\|$ represents a mismatch between the reference model dynamics and the multiple agent controlled system; the result in the

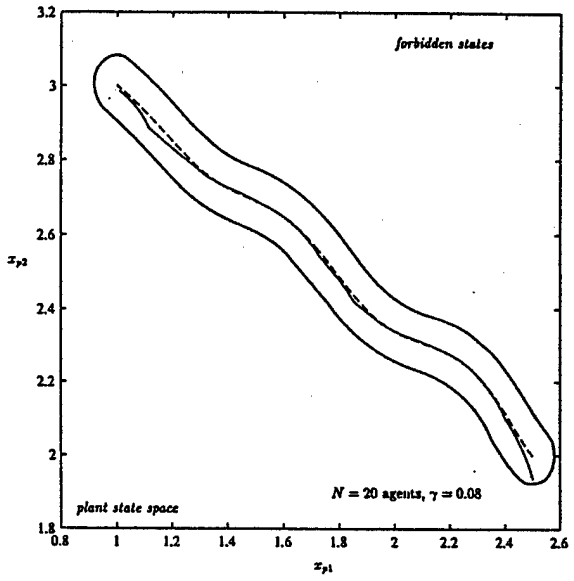


Figure 3: Simulation for 20 agents with $\gamma = 0.08$. Reference trajectory (dashed) and controlled plant state (solid) with forbidden regions (shaded) are shown.

figure indicates that an increase in the number of agents (via a reduction in ϑ) results in improved dynamical model matching. The other performance curves are self-explanatory.

As with the previous example of the LP-approach, this example depicts some of the fundamental characteristics of the MRC-approach. The approach is an off-line procedure which requires the solution of LMI problems. In the present form, the computation of the control requires explicit knowledge of the plant dynamics and direct measurement of the plant state. However, because the approach is based primarily on Lyapunov and structured uncertainty methods for robust control design, the approach should be extendable to uncertain systems. The computational burden is large, but it is off-line and the payoff is a non-chattering control which satisfies amplitude constraints.

The MRC approach is a new application of classical gain scheduling and robust control techniques in the following respects. First, classical gain scheduling offers no systematic checks for stability and performance in a bounded amplitude performance problem; those results which appear in the literature (e.g., [Shamma 1990][Shamma 1991]) concern

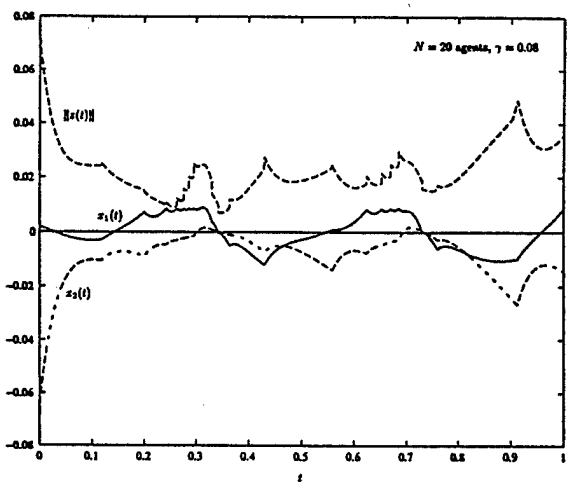


Figure 4: Simulation for 20 agents with $\gamma = 0.08$. Error states x_1 and x_2 with performance $\|z(t)\|$ are shown.

bounded energy ($\mathcal{L}_2$) performance problems. A similar claim is true for robust control techniques which almost exclusively apply to $\mathcal{L}_2$ problems. In addition, robust control techniques do not apply in a direct manner to switched-agent control problems such as the one considered here. The MRC method represents a combination of the two techniques for bounded-amplitude problems which arise naturally in hybrid system applications.

5 Conclusions

This paper has compared two methods for safe implementation of supervisory commands in hybrid dynamical control systems, called the LP method and the MRC method. Both methods appear to be able to guarantee the bounded amplitude performance requirements dictated by the hybrid design problem, assuming knowledge of the plant dynamics. The LP method produces a chattering control policy versus the non-chattering control policy generated by the MRC method. Both methods require that the plant dynamics do not vary too rapidly.

As presented, both methods require knowledge of the plant dynamics and full state

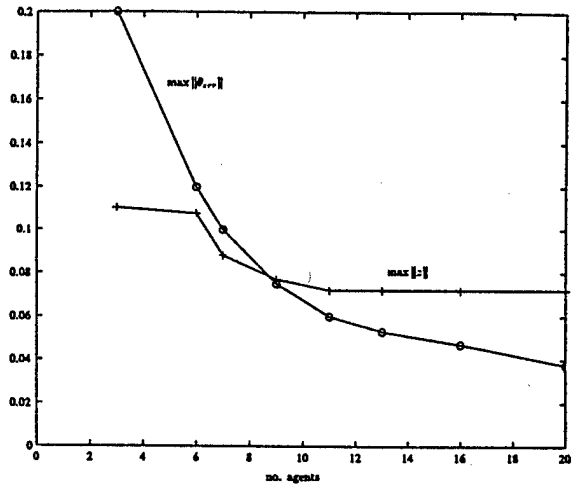


Figure 5: Average performance versus number of agents

availability. This is required in order to compare the two approaches. While it is unclear if this assumption may be relaxed in the LP method, the MRC method can be extended to structurally perturbed systems and output feedback cases in a straightforward fashion because it is based primarily on linear robust control and Lyapunov techniques. (This is a topic of current research efforts.) While the extension is straightforward, it is not trivial and adds considerable complexity to the presentation; it is not included in this paper. We note that the underlying structure of the MRC method allows the method to be generalized in another direction, as well; namely, more complex control agents may be used. The most obvious extension is to bounded amplitude LPV control agents, analogous to those discussed in [Packard 1994].

To emphasize, while both methods require explicit knowledge of the plant dynamics, the MRC method appears to be more amenable to incorporation of modeling uncertainty and disturbances into the design, yielding robust control policies. Furthermore, the designs may be accomplished using the same tools as for the nominal case since the design tools are linear robust control techniques. The LP method may offer such advantages, but they are not apparent.

In the area of numerical complexity, the LP method requires the solution of simple linear programming problems which, of course, can be solved quickly and efficiently. This advantage is offset, somewhat, by the fact that the linear programs must be solved on-line and often. On the other hand, the MRC method requires the solution of a series of larger convex optimization problems. However, while this requires a more computationally intensive effort, the procedure is performed off-line and must only be performed once.

References

[Bett 1997] C. Bett and M. Lemmon. Finite-horizon induced-$\mathcal{L}_\infty$ performance of linear parameter varying systems. In *Proceedings of the American Control Conference*, Albuquerque, NM, June 1997. Also appears as Technical Report ISIS-97-002, Department of Electrical Engineering, University of Notre Dame, Notre Dame, IN, 1997.

[Boyd 1994] Stephen Boyd, Laurent El Gaoui, Eric Feron, and Venkataramanan Balakrishnan. *Linear Matrix Inequalities in System and Control Theory.* Society for Industrial and Applied Mathematics, 1994.

[Dahleh 1987] M.A. Dahleh and J.B. Pearson. L^1-optimal compensators for continuous-time systems. *IEEE Transactions on Automatic Control*, 32(10):889–895, October 1987.

[Deshpande 1995] Akash Deshpande and Pravin Varaiya. Viable control of hybrid systems. In Panos Antsaklis, Wolf Kohn, Anil Nerode, and Shankar Sastry, editors, *Hybrid Systems II*, pages 128–147. Springer-Verlag, 1995. Lecture Notes in Computer Science, Volume 999.

[Isidori 1989] A. Isidori. *Nonlinear Control Systems.* Springer-Verlag, 2nd edition, 1989.

[Kohn 1993] W. Kohn and A. Nerode. Multiple agent autonomous hybrid control systems. In Grossman, Nerode, Ravn, and Rischel, editors, *Hybrid Systems*, pages 297–316. Springer-Verlag, 1993. Lecture Notes in Computer Science, Volume 736.

[Lemmon 1996] M. Lemmon and C. Bett. Robust hybrid control system design. In *Proceedings of the 1996 IFAC World Congress*, volume J, pages 395–400, San Francisco, CA, June 1996.

[Nagpal 1994] K. Nagpal, J. Abedor, and K. Poolla. An LMI approach to peak-to-peak gain minimization: filtering and control. In *Proceedings of the American Control Conference*, pages 742–746, Baltimore, Maryland, June 1994.

[Packard 1994] Andy Packard. Gain scheduling via linear fractional transformations. *Systems and Control Letters*, 22:79–92, 1994.

[Shamma 1990] Jeff S. Shamma and Michael Athans. Analysis of gain scheduled control for nonlinear plants. *IEEE Transactions on Automatic Control*, 35(8):898–907, 1990.

[Shamma 1991] Jeff S. Shamma and Michael Athans. Guaranteed properties of gain scheduled control for linear parameter-varying plants. *Automatica*, 27(3):559–564, 1991.

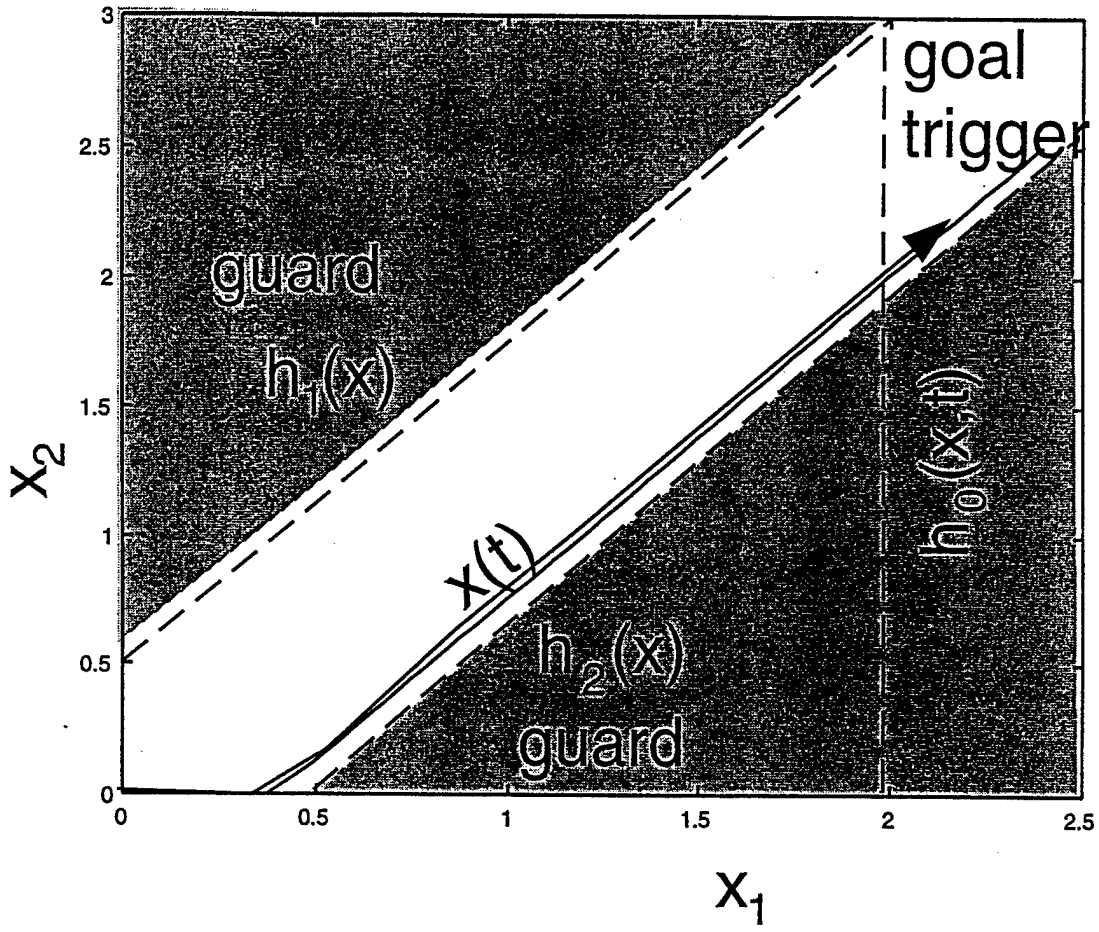


Figure 1: Guard and goal triggers for example.

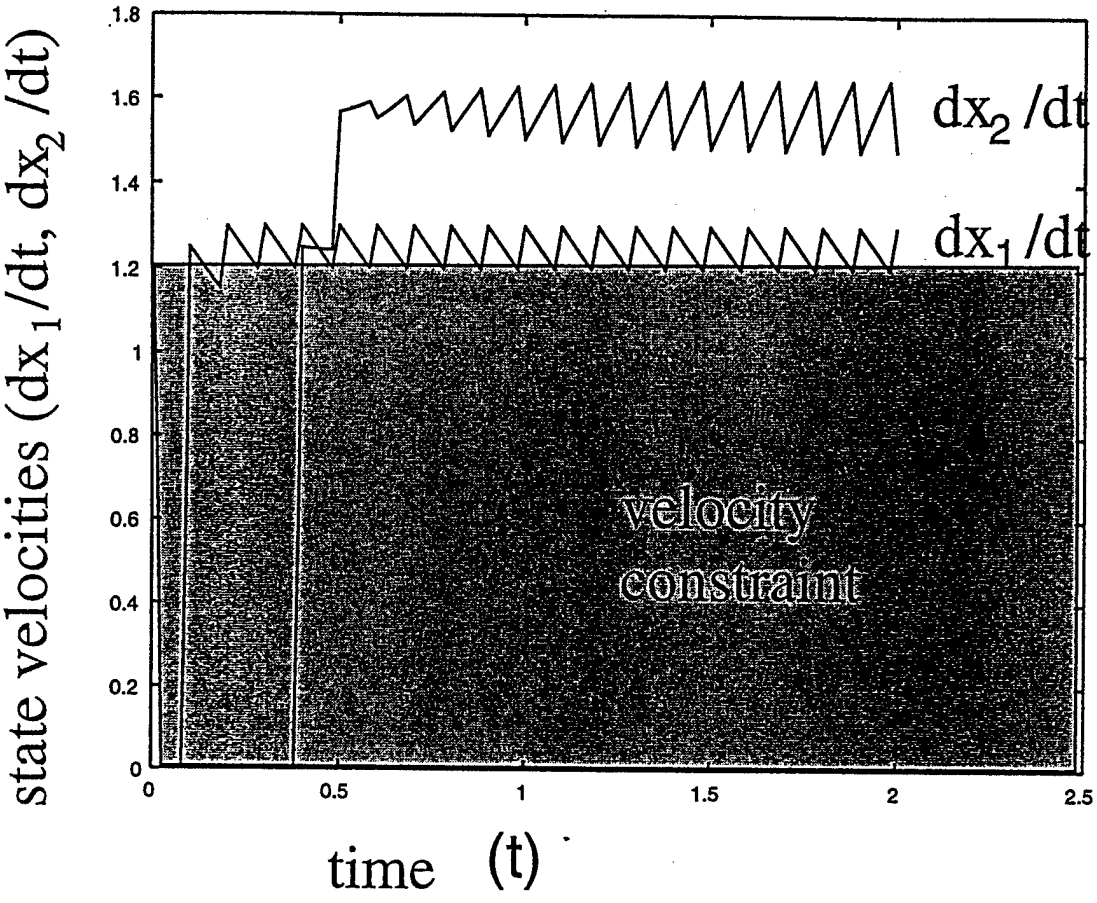


Figure 2: Chattering control policy

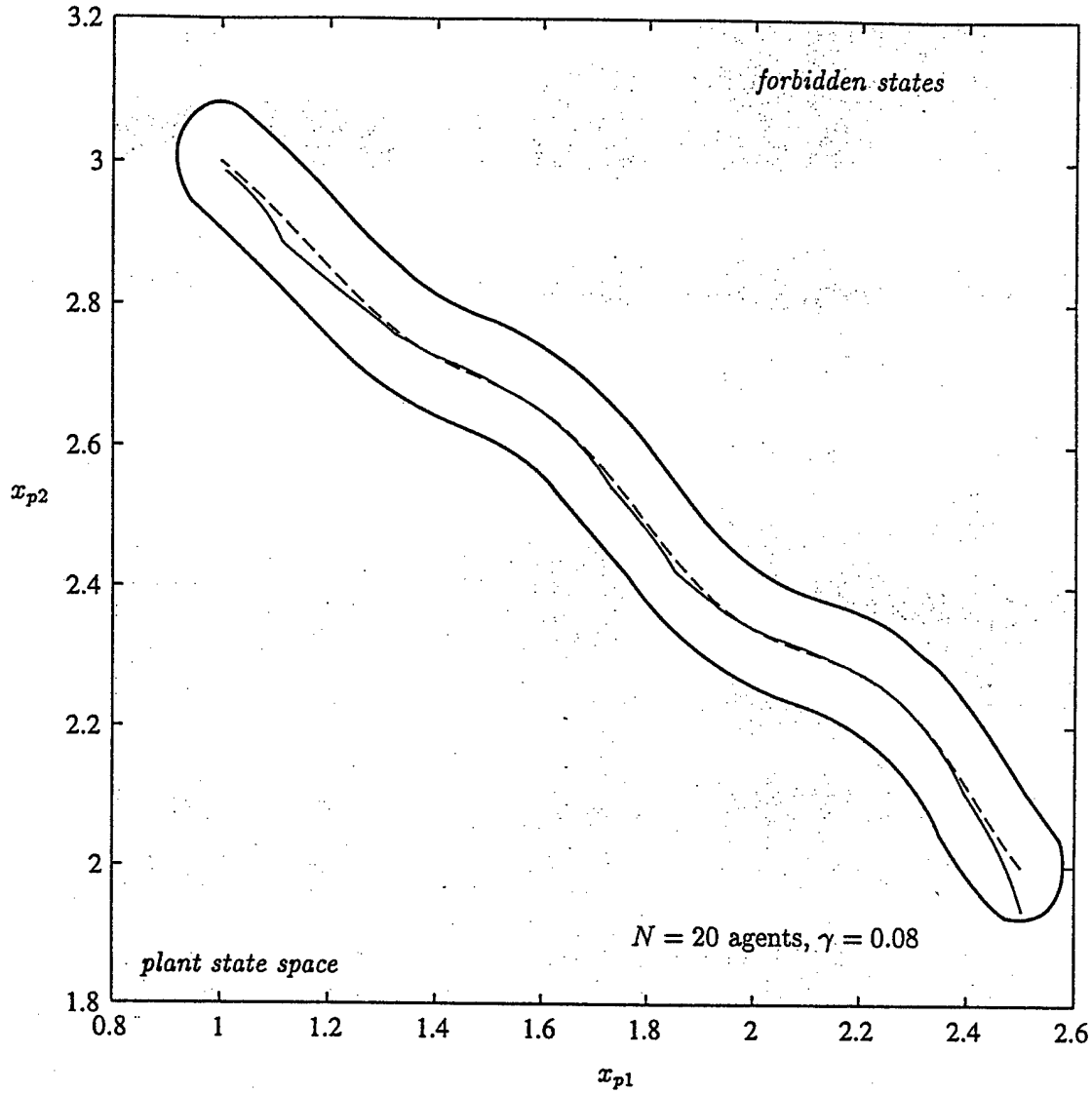


Figure 3: Simulation for 20 agents with $\gamma = 0.08$. Reference trajectory (dashed) and controlled plant state (solid) with forbidden regions (shaded) are shown.

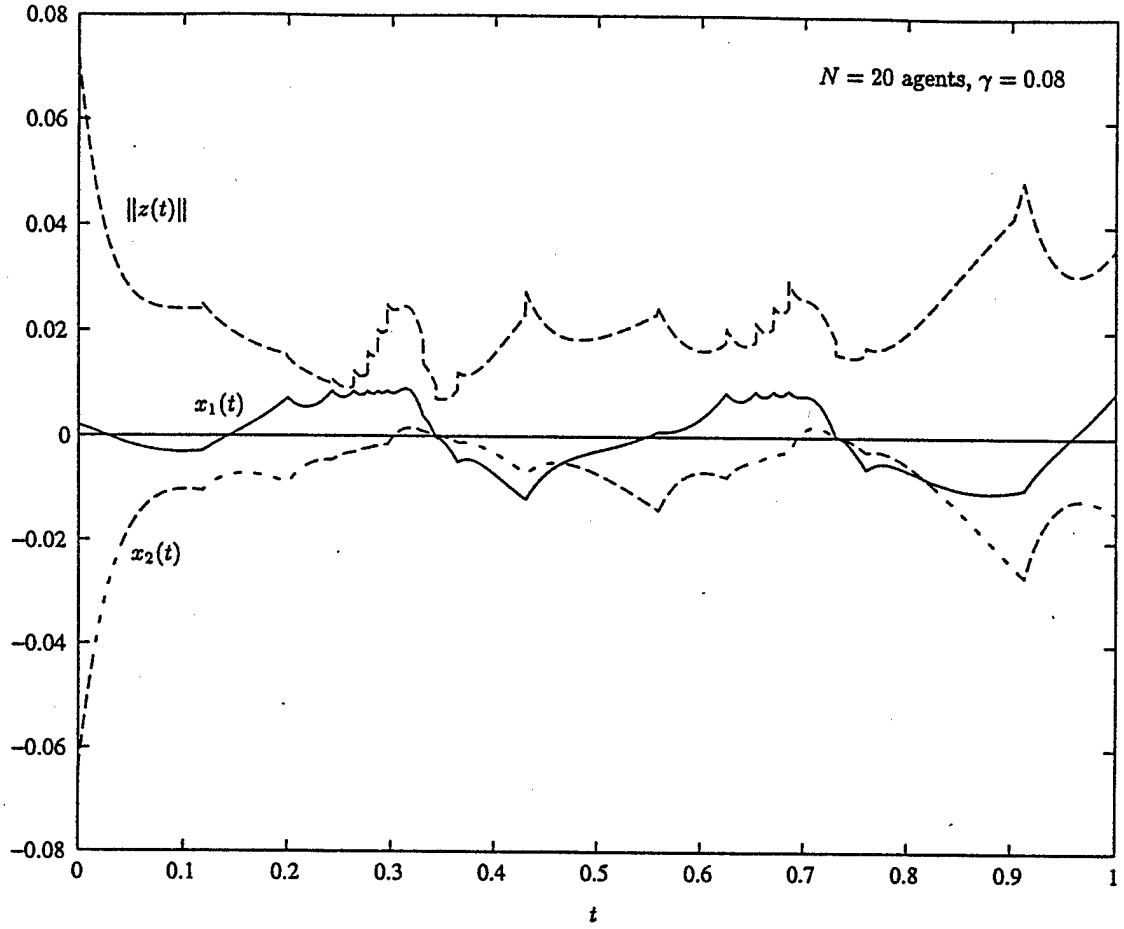


Figure 4: Simulation for 20 agents with $\gamma = 0.08$. Error states x_1 and x_2 with performance $\|z(t)\|$ are shown.

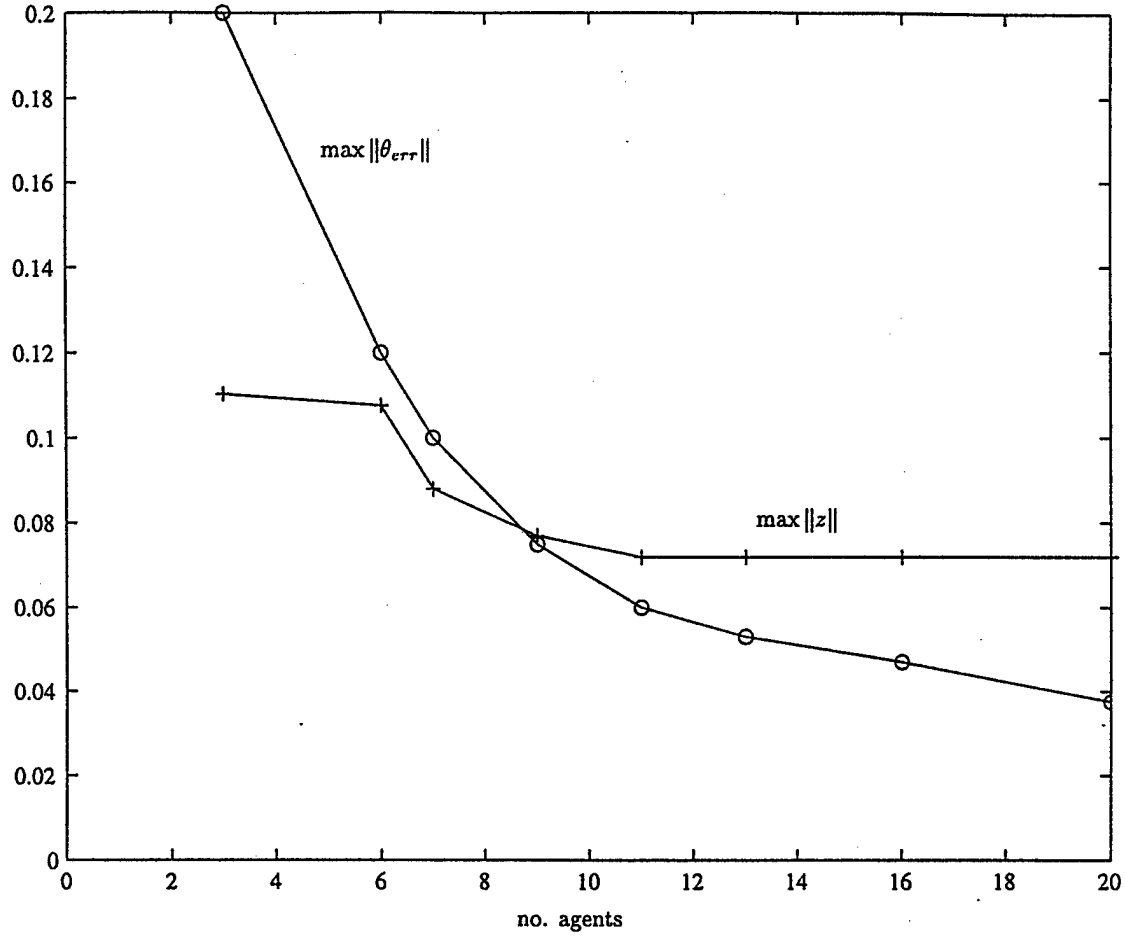


Figure 5: Average performance versus number of agents

APPENDIX D

C.J. Bett and M.D. Lemmon
"Bounded Amplitude Performance of Switched LPV Systems with Applications to Hybrid Systems"
Submitted to *Automatica* special issue on hybrid systems, September 1997
Expanded version of technical report ISIS-97-004
University of Notre Dame, March 1997.

Bounded Amplitude Performance of Switched LPV Systems with Applications to Hybrid Systems

Christopher J. Bett and Michael D. Lemmon*

Department of Electrical Engineering
University of Notre Dame
Notre Dame, IN 46556

September 1997

Abstract

This paper discusses recent results on multiple linear agent control for systems satisfying a bounded amplitude performance constraint. The plant is assumed to be a linear parameter varying (LPV) system scheduled along a nominal parameter trajectory; in this respect, the control problem represents a plant operating between a prespecified set of operating conditions. Linear controllers are designed at setpoints along this scheduling trajectory to satisfy bounded amplitude performance constraints. This paper discusses an approach to analyze the switched system behavior under practical assumptions on the structure of the switching rule. The approach combines the scheduling parameter with LPV system properties to derive bounds on the switching behavior of the system. These estimates are then used to construct a logical model of the switched system behavior in the form of a timed automaton. In this respect, this paper presents a way of extracting logical models of continuous time system behavior.

*The authors gratefully acknowledge the partial financial support of the Army Research Office (DAAH04-95-1-0600, DAAH04-96-1-0134).

1 Introduction

Linear feedback control assumes that the operating range of the plant is sufficiently small to allow the use of linear approximations of the plant dynamics in designing feedback control laws. In cases where the operating region is so large that linear controllers are unable to meet performance requirements, it is necessary to use nonlinear control techniques. A widely used approach is *multiple agent control.* This method uses a finite collection of feedback controllers (called *control agents*) to achieve a specified performance level. Outputs of the control agents are combined according to some function of the plant state and used to control the plant. In typical gain scheduling methods, a weighted sum of control agent outputs is used as the control input. If these weights are chosen either 1 or 0 (i.e., either "on" or "off"), then a *switched agent control system* is obtained.

In this paper, the plant is assumed to be a linear parameter varying (LPV) system. The control objective is to maintain an amplitude performance constraint by switching among a set of LTI controllers while the LPV system is scheduled along a known reference parameter trajectory. The LPV problem formulation is useful for modeling a large class of nonlinear control problems including classical gain scheduling and nonlinear model reference control problems. The bounded amplitude performance objective arises naturally in many applications, including robotics, high performance drive positioning (e.g., disk drives) and hybrid system applications where it is useful to avoid certain regions in the state space.

Building on prior results characterizing linear controllers of non-switched LPV systems for bounded amplitude performance [5], the results of this paper show how Lyapunov and robust control techniques for structured perturbations to linear systems may be used to derive sufficient conditions for the switched system to meet bounded amplitude performance requirements. These conditions, which are the primary contribution of the paper, amount to ensuring that a *dwell-time* condition be satisfied by each control agent which is switched into the feedback loop. These dwell-time conditions are directly related to a uniform ultimate boundedness condition (see, e.g. [14]) on the controlled system.

In the switched system context just described, the concept of dwell-time appears to have been introduced in [17]. There, linear control agents are employed and switched into feedback according to a prediction of controller performance. The results presented here are distinct form those in [17] and more recent work on switched system stability [12] in that those results concern an $\mathcal{L}_2$ performance criteria, not the bounded amplitude constraint stated above. Nonlinear control agents are used for bounded amplitude performance in [18]. In that work, the authors explore the effects of switching transients and derive bounds on these

transients. Unlike [18], the results presented here can be used to synthesize agents to meet prespecified amplitude constraints.

The switched agent control systems described in this paper and in [17][18][12] all fall into the class of switched systems described in [8]. Such systems are often referred to as *hybrid systems* because they generate a mixture of both discrete event and continuous-valued signals. As noted earlier, we describe the performance of hybrid systems using a bounded amplitude performance measure. Such measures are well-suited to hybrid systems since discrete events are often defined in terms of set boundaries, or guards, in the continuous system state space. Avoiding such boundaries is an amplitude control problem. In general, analysis of such systems for stability and performance is difficult because of the mix of continuous and discrete behavior. The results in [17][18][12] all approach the analysis problem in the continuous domain.

The other obvious approach is to analyze the hybrid system is from the discrete dynamical point of view. One extreme example of this approach is found in [22] where a continuous dynamical system is abstracted into an untimed finite automata. Another approach, proposed in [2], abstracts the continuous dynamics to a simple linear differential inclusion and then a timed-automaton. Such approaches offer enormous computational advantages over the continuous-time counterparts described above and can be applied, in some cases, to systems with a large number of states. However, the abstraction process often ignores the inherent structure of the underlying continuous-time dynamical system, often ignoring important questions of stability and performance in the continuous domain. As a result, much additional computation and analysis is often required.

The development of a useful and efficient design methodology for hybrid system design and analysis requires the integration of the continuous domain approach and the discrete event approach. While the primary results of this paper analyze the switched system in the continuous domain, the analysis tools offer a means of obtaining a timed logical model of the system in the form of a timed-automaton model of the switched system behavior. This is the second contribution of the paper. The ability to perform such an abstraction is useful because it allows one to analyze distinctly different systems exhibiting timed logical behavior in the logical domain. In the case of complex systems consisting of smaller subsystems which must be coordinated in some fashion, it allows one to extract the timed-logical models for each of the subsystems. The logical models may then be used to verify whether or not the supervised system meets timing specifications expressed as temporal logic formulae.

The remainder of this paper is organized as follows. Mathematical background is summarized in section

2. The problem setup, including assumptions on the plant and control agents are described in section 3. Section 4 states sufficient conditions for the bounded amplitude performance of a continuous-time system whose dynamics switch between two different LPV realizations. These conditions provide guidelines for the analysis presented in section 5. Section 5 contains the conditions for estimating the switching times of the scheduled system which can then be compared to a *dwell-time* condition to check for performance and stability of the switched system. Section 6 demonstrates, by example, how the continuous-time results may be used to extract timed-automata models of a simple process control system. Both finite-time and periodic scheduling examples are provided. All proofs are located in appendix A.

2 Preliminaries

This section establishes the mathematical notation used throughout the paper.

Definition 2.1 *For a finite constant $T > 0$, the finite-horizon infinity norm of a signal $f : \mathbb{R}^+ \to \mathbb{R}^n$ is defined as*

$$\|f\|_{\infty,[0,T]} := \operatorname{ess\,sup}_{t\in[0,T]} \|f(t)\|$$

where $\|\cdot\|$ denotes the Euclidean vector norm. The linear space $\mathcal{L}^n_\infty[0,T]$ is defined by

$$\mathcal{L}^n_\infty[0,T] := \left\{ f : \mathbb{R}^+ \to \mathbb{R}^n \,\middle|\, \|f\|_{\infty,[0,T]} < \infty \right\}$$

The subset $\left\{ f : \mathbb{R}^+ \to \mathbb{R}^n \,\middle|\, \|f\|_{\infty,[0,T]} \leq 1 \right\} \subset \mathcal{L}^n_\infty[0,T]$ is denoted $\mathcal{BL}^n_\infty[0,T]$.

The infinite-horizon infinity norm of a signal $f : \mathbb{R}^+ \to \mathbb{R}^n$ is defined as

$$\|f\|_{\infty,[0,\infty)} := \operatorname{ess\,sup}_{t\in[0,\infty)} \|f(t)\|$$

where $\|\cdot\|$ denotes the Euclidean l_2 vector norm. The linear space $\mathcal{L}^n_\infty[0,\infty)$ is defined by

$$\mathcal{L}^n_\infty[0,\infty) := \left\{ f : \mathbb{R}^+ \to \mathbb{R}^n \,\middle|\, \|f\|_{\infty,[0,\infty)} < \infty \right\}$$

The subset $\left\{ f : \mathbb{R}^+ \to \mathbb{R}^n \,\middle|\, \|f\|_{\infty,[0,\infty)} \leq 1 \right\} \subset \mathcal{L}^n_\infty[0,\infty)$ is denoted $\mathcal{BL}^n_\infty[0,\infty)$. The spaces $\mathcal{L}^n_\infty[0,\infty)$ and $\mathcal{BL}^n_\infty[0,\infty)$ will often be denoted, respectively, $\mathcal{L}^n_\infty$ and $\mathcal{BL}^n_\infty$.

Definition 2.2 (Parameter Variation Set) *Given a compact subset $\tilde{\Theta} \subset \mathbb{R}^s$, the* parameter variation set *$\mathcal{F}_{\tilde{\Theta}}$ denotes the set of all continuous functions mapping $\mathbb{R}^+$ into $\tilde{\Theta}$. For a finite $T > 0$ and a compact subset $\Theta \subseteq \tilde{\Theta}$, the set $\mathcal{F}_\Theta[0,T]$ denotes the set of all continuous functions $\theta \in \mathcal{F}_{\tilde{\Theta}}$ which map $[0,T]$ into Θ.*

The notation $\theta \in \mathcal{F}_{\bar{\Theta}}$ denotes a function in the parameter variation set; $\theta \in \bar{\Theta}$ denotes a vector in a compact subset of $\mathbb{R}^s$. These definitions are extended to functions which map finite time intervals to compact subsets of $\bar{\Theta}$. It is clear that $\mathcal{F}_\Theta \subset \mathcal{F}_\Theta[0,T] \subset \mathcal{F}_{\bar{\Theta}}$.

Note that both $\mathcal{F}_{\bar{\Theta}}$ and, for instance, $\mathcal{L}^n_\infty$ represent signal spaces. Technically, it can easily be argued that $\mathcal{F}_{\bar{\Theta}} \subset \mathcal{L}^s_\infty$ since $\mathcal{F}_{\bar{\Theta}}$ consists of supremum bounded s-dimensional vectors which vary continuously in time. In this paper, the following convention is followed: $\mathcal{F}_{\bar{\Theta}}$ will always refer to parameter signals or parameter variations; $\mathcal{L}^n_\infty$ will refer to signals in the plant input, output or state space.

Definition 2.3 (Linear Parameter Varying (LPV) System) *Given a compact set $\bar{\Theta} \subset \mathbb{R}^s$, and continuous functions $A : \mathbb{R}^s \to \mathbb{R}^{n\times n}$, $B : \mathbb{R}^s \to \mathbb{R}^{n\times n_w}$, $C : \mathbb{R}^s \to \mathbb{R}^{n_z\times n}$, and $D : \mathbb{R}^s \to \mathbb{R}^{n_z\times n_w}$, an n^{th} order linear parameter varying (LPV) system is a dynamical system whose dynamics evolve as*

$$\begin{bmatrix} \dot{x}(t) \\ z(t) \end{bmatrix} = \begin{bmatrix} A(\theta(t)) & B(\theta(t)) \\ C(\theta(t)) & D(\theta(t)) \end{bmatrix} \begin{bmatrix} x(t) \\ w(t) \end{bmatrix} \tag{1}$$

where $\theta \in \mathcal{F}_{\bar{\Theta}}$.

Note that the LPV system is defined over the parameter variation set $\mathcal{F}_{\bar{\Theta}}$. Certain properties of the LPV system may only be guaranteed while the parameter variations are confined to a subset of $\mathcal{F}_{\bar{\Theta}}$. Previous results on LPV systems (see, for example, [4], [20], [13]) restrict consideration to properties which can be guaranteed over the entire parameter set $\bar{\Theta}$ and the associated parameter variation set. This restricts the class of systems for which the linear design techniques may be applied to systems where $\bar{\Theta}$ is small or possesses sufficient structure in the parameter dependence.

Definition 2.4 *Let the LPV system of definition 2.3 be denoted by $\Sigma(\bar{\Theta}, A, B, C, D)$. Then for any $\theta \in \mathcal{F}_{\bar{\Theta}}$,*

- *the linear time-varying system described in equation 1 is denoted Σ_θ,*
- *$\Phi_\theta(t, t_0)$ is the state-transition matrix of of Σ_θ*
- *for $x(t_0) = 0$, the causal linear input/output mapping, $H_\theta : \mathcal{L}^{n_w}_\infty \to \mathcal{L}^{n_z}_\infty$, of Σ_θ is defined as*

$$H_\theta w(t) = \int_{t_0}^{t} C(\theta(\tau))\Phi_\theta(t,\tau)B(\theta(\tau))w(\tau)d\tau + D(\theta(t))w(t)$$

- *for a finite $T > 0$, the finite-horizon induced-$\mathcal{L}_\infty$ norm of H_θ is given by*

$$\|H_\theta\|_{i\infty,[0,T]} := \sup_{w\in\mathcal{B}\mathcal{L}^{n_w}_\infty[0,T]} \|H_\theta w\|_{\infty,[0,T]} \tag{2}$$

- *the infinite-horizon induced-$\mathcal{L}_\infty$ norm of H_θ is given by*

$$\|H_\theta\|_{i\infty} := \sup_{w \in B\mathcal{L}_\infty^{n_w}} \|H_\theta w\|_\infty \tag{3}$$

The set $H_{\mathcal{F}_{\tilde{\Theta}}}$ is defined as

$$H_{\mathcal{F}_{\tilde{\Theta}}} := \{H_\theta : \theta \in \mathcal{F}_{\tilde{\Theta}}\}$$

The shorthand $\|H_{\mathcal{F}_{\tilde{\Theta}}}\|_{i\infty} \leq \gamma$ for some $\gamma > 0$ means that for all $\theta \in \mathcal{F}_{\tilde{\Theta}}$, $\|H_\theta\|_{i\infty} \leq \gamma$.

Finally, a positive definite matrix $P \in \mathbb{R}^{n \times n}$ which satisfies

$$A'P + PA + \alpha_1 P + \frac{1}{\alpha_2} PB'BP \leq 0 \tag{4}$$

for $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n_u \times n}$, and scalar α_1 and α_2, will be denoted

$$P \in \text{FeasRic}\,(A, B, \alpha_1, \alpha_2)$$

3 Problem Description

In this section, the control system architecture is described and the performance objective for the controlled system is stated. The components of the system architecture are depicted in figure 1.

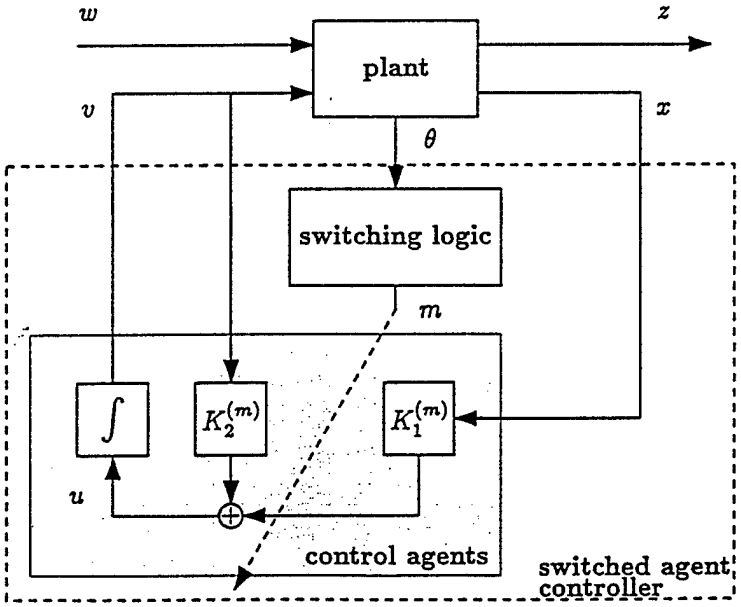


Figure 1: Control System Architecture

3.1 Plant Dynamics

The plant processes considered in this paper are assumed to take the form of an LPV system

$$\dot{x}(t) = A_1(\theta(t))x(t) + B_1(\theta(t))w(t) + B_2(\theta(t))v(t) \tag{5}$$

$$z(t) = C_1 x(t) + D_{12} v(t) \tag{6}$$

with $\theta \in \mathcal{F}_{\tilde{\Theta}}$ where $\tilde{\Theta}$ is some compact subset of $\mathbb{R}^s$. Here, $x \in \mathbb{R}^n$ is the plant state, $v \in \mathbb{R}^{n_v}$ is the control input, $w \in \mathbb{R}^{n_w}$ represents bounded, exogenous disturbances and $z \in \mathbb{R}^{n_z}$ represents plant performance. It is assumed that the disturbance vector $w \in \mathcal{BL}_\infty^{n_w}$.

The parameter vector θ is assumed to be defined by a mapping $S : \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^{n_v} \to \mathbb{R}^s$, called a *parameter mapping*, so that $\theta(t) = S(x_m(t), x(t), v(t))$; the argument $x_m \in \mathbb{R}^n$ represents an exogenous reference or scheduling variable. While the specific form of S is not central to results of this paper, it is assumed that S is continuous, available for measurement and bounded so that there exists a known constant $k_{zi} > 0$ such that for any $x_m \in \mathbb{R}^n$, $x_1, x_2 \in \mathbb{R}^n$, and $v_1, v_2 \in \mathbb{R}^{n_v}$

$$|S_i(x_m, x_1, v_1) - S_i(x_m, x_2, v_2)| \leq k_{zi} \|z_1 - z_2\| \tag{7}$$

where $\|\cdot\|$ denotes the Euclidean vector norm, $S_i(\cdot,\cdot,\cdot)$ is the ith element of the parameter vector and $z_j = C_1 x_j + D_{12} v_j$ for $j = 1, 2$.

Remark: Systems of the form described above arise naturally in many applications. When the parameter mapping is purely a function of exogenous parameters, x_m, which are available for measurement, then the system describes a class of LPV gain scheduling problems (see, e.g. [21][20][3]). Similarly, when θ depends on the system state, the LPV systems which result are sometimes referred to as *quasi-LPV* systems. (Such systems have been studied extensively in, for example, [19].) LPV (or quasi-LPV) systems arise in nonlinear model reference control problems where ideal performance is measured in terms of a dynamical reference model with state, x_m, and state error, x. A parameter mapping arises from grouping state and control dependent terms in the coefficient matrices of the system.

The evolution of $\theta(t)$ in $\tilde{\Theta} \subset \mathbb{R}^s$ will be called the *parameter trajectory*. We will assume the existence of a *known* parameter trajectory representing ideal performance; this special parameter trajectory will be called the *nominal parameter trajectory* defined as

$$\theta_{nom}(t) := S(x_m(t), 0, v_m(t)) \tag{8}$$

The argument $v_m(t)$ represents a nominal control input to the plant. In some cases, such a function may

be derived analytically from knowledge of the plant dynamics and the control objectives, e.g. a feedback linearizing control. (Implementation of the feedback linearizing control may not be desirable for robustness reasons or if hardware constraints do not permit.) In other cases, $v_m(t)$ may be found by computing local solutions to linearized problems and interpolating the results.

Remark: The nominal parameter trajectory is said to represent ideal performance because it corresponds to the reference or scheduling variable, x_m, and a control, v_m, which is assumed to work well in the ideal case. "Ideal" performance can be interpreted in another way. Note that freezing θ at a point and evaluation the LPV plant results in an LTI system. One may think of such a process along points of the nominal parameter trajectory as generating a family of nominal LTI plants. Deviations of the parameter trajectory, $\theta(t)$, from the nominal parameter trajectory may thus be treated as perturbations to the nominal system. •

Finally, without loss of generality, assume that for any points $\theta \in \tilde{\Theta}$ that

$$\text{rank}\left[B_2(\theta)|A_1(\theta)B_2(\theta)|A_1^2(\theta)B_2(\theta)|\cdots|A_1^{n-1}(\theta)B_2(\theta)\right] = n \tag{9}$$

This assumption implies that LTI plant models for all frozen values of $\theta \in \tilde{\Theta}$ yield a controllable linear system.

3.2 Performance Objective

As noted earlier, attention in this paper is restricted to *bounded amplitude performance problems.* Let γ be a fixed positive constant representing a performance level and let $\|\cdot\|$ indicates the vector 2-norm. Two types of performance problems will be considered in this paper

Finite-time scheduling Consider the LPV system described in (5-6). The finite-time bounded amplitude performance objective is to ensure that, given $\|z(0)\| \leq \gamma$,

$$\sup_{w \in \mathcal{BL}_\infty^{n_w}[0,T]} \|z(t)\| \leq \gamma \tag{10}$$

where T is a fixed positive constants.

Periodic scheduling Consider the LPV system described in (5-6) with a periodic nominal parameter trajectory,

$$\theta_{nom}(t) = \theta_{nom}(t+T)$$

The bounded amplitude performance objective in this case is to ensure that, given $\|z(0)\| \leq \gamma$,

$$\sup_{w \in \mathcal{BL}_\infty^{n_w}} \|z(t)\| \leq \gamma. \tag{11}$$

Both of these objectives represent important classes of performance problems. Specifically, the first represents tasks which reach completion in a finite time. The second represents cyclic processes which run continually. The purpose of this paper is analyze these performance problems for the class of plant processes just described under a control of a class of switched agent controllers, as depicted in figure 1, which are described next.

3.3 Switched Agent Controller Structure

This subsection describes the components of the switched agent controller assumed for the LPV system described above and depicted in figure 1. There are two primary components to the switched agent controller: control agents and switching logic. These are now described.

Control Agents Consider the LPV system described in (5-6). Consider a sequence of times, $\{t_i\}$ indexed by $i \in \mathcal{I}_{\mathcal{K}} = \{1, \ldots, M\}$. The parameter vectors obtained by sampling the nominal parameter trajectory at times t_i for $i \in \mathcal{I}_{\mathcal{K}}$ form a finite collection of *design points*. In other words,

$$\text{Design Points} = \{\theta : \theta = \theta_{nom}(t_i), i \in \mathcal{I}_{\mathcal{K}}\} \tag{12}$$

The ith design point will be denoted as $\theta_{nom}^{(i)}$. With each $i \in \mathcal{I}_{\mathcal{K}}$, associate a *control agent* designed for the LPV system when the parameter is fixed at the design point $\theta_{nom}^{(i)}$. The ith control agent will be represented by the system

$$\dot{v} = u \tag{13}$$

$$u = K_1^{(i)} x + K_2^{(i)} v \tag{14}$$

where $K_1^{(i)}$ and $K_2^{(i)}$ are constant gain matrices of appropriate dimensions. The collection of control agents, for $i \in \mathcal{I}_{\mathcal{K}}$ will be denoted $\mathcal{K}$.

For analysis and synthesis purposes, the integrator in the control agent will be incorporated into the plant. The modified LPV plant is given by

$$\dot{\hat{x}}(t) = A(\theta(t))\hat{x}(t) + B_w(\theta(t))w(t) + B_u u(t) \tag{15}$$

$$z(t) = C\hat{x}(t) \tag{16}$$

where

$$A(\theta(t)) = \begin{bmatrix} A_1(\theta(t)) & B_2(\theta(t)) \\ 0 & 0 \end{bmatrix}, B_w(\theta(t)) = \begin{bmatrix} B_1(\theta(t)) \\ 0 \end{bmatrix}, B_u = \begin{bmatrix} 0 \\ I \end{bmatrix}, C = \begin{bmatrix} C_1 & D_{12} \end{bmatrix} \tag{17}$$

and $\hat{x} = \begin{bmatrix} x' & v' \end{bmatrix}$. The modified plant is now seen to have a control agent $K^{(i)} = \begin{bmatrix} K_1^{(i)} & K_2^{(i)} \end{bmatrix}$ so that $u = K^{(i)}\hat{x}$.

Switching Logic: The switching logic is the set of rules which define how the control agents are switched into feedback with the plant process. In this paper, the switching logic is assumed to have two components called the *switching sets* and the *nearest neighbor switching rule.* These components are now described.

Switching Sets: Switching between the different control agents in $\mathcal{K}$ will be controlled by the parameter vector, θ. In particular, associate with each element of $\mathcal{K}$ a compact subset of the parameter set $\tilde{\Theta}$. This set will be called the *switching set*; the ith switching set associated with control agent $K^{(i)}$ will be denoted as $\Theta_i \subset \tilde{\Theta}$. While this set can be chosen in many ways, attention in this paper is confined to switching sets of the form

$$\Theta_i := \left\{ \theta \;\middle|\; \max_j \left| \theta_j - \theta^{(i)}_{nom,j} \right| \leq \vartheta_{out} \right\} \tag{18}$$

where θ_j and $\theta^{(i)}_{nom,j}$ denote the jth components of the parameter vectors θ and $\theta^{(i)}_{nom}$, respectively, and ϑ_{out} is a parameter quantifying the size of the switching set. ϑ_{out} will also be referred to as the *switching parameter.*

Nearest-Neighbor Switching Rule: For a given collection of control agents $\mathcal{K}$ with associated switching sets, there are a variety of *switching rules* which can be invoked. In this paper attention is focused on a *nearest neighbor* switching rule. Given the switching parameter ϑ_{out}, a collection of control agents $\mathcal{K}$, and a collection of parameter sets $\mathcal{C} = \{\Theta_i\}$ as defined in equation 18, suppose that control agent $K^{(i)}$ is in the feedback loop at time t_0 and assume that $\theta(t_0) \in \Theta_i$. Then the control agent $K^{(i)}$ will remain in the feedback loop until the earliest time t_s when the parameter trajectory $\theta(t)$ satisfies

$$\max_j \left| \theta_j(t_s) - \theta^{(i)}_{nom,j} \right| = \vartheta_{out} \tag{19}$$

At time t_s, the control agent $K^{(m)}$ is then switched into the feedback loop where

$$m = \arg \min_{j \in \mathcal{I}_{\mathcal{K}}} \left\| \theta(t_s) - \theta^{(j)}_{nom} \right\| \tag{20}$$

where $\| \cdot \|$ denotes the Euclidean 2-norm.

As described, the switched agent controller switches state feedback controllers into and out of the feedback loop on the basis of the LPV system's current parameter vector. In order to have a well-behaved switched system, it is necessary that the parameter vector after a switch lie in the switching sets associated with the control agent which is currently in the feedback loop. The parameter trajectory is called *legal* if this occurs. More precisely, let $\mathcal{C}$ be the collection of switching sets. A parameter trajectory $\theta(t)$ will be said to be *legal* if and only if it is continuous (except possibly at switching instants) and $\theta(t) \in \Theta_i$ for all $t \in \left\{\tau : u(\tau) = K_1^{(i)}x(\tau) + K_2^{(i)}v(\tau)\right\}$. In particular, a legal parameter trajectory is denoted $\theta \in \mathcal{F}_{\mathcal{C}}$.

Adequate Sampling Assumption:

From the preceding discussion it is clear a given control agent, $K^{(i)}$, is switched out of the feedback loop when the parameter $\theta(t)$ leaves the switching set Θ_i. The nearest neighbor switching rule says that the resulting switch will be to the controller, $K^{(j)}$, whose associated design point $\theta_{nom}^{(j)}$ is closest (with respect to the Euclidean vector norm) to the parameter $\theta(t_s)$ at switching time t_s. In order to guarantee performance properties, it would be advantageous if the parameter $\theta(t_s^+)$ immediately after the switch were in the switching set Θ_j. To help guarantee this property, it will be assumed that for all t there exists $l \in \mathcal{I}_{\mathcal{K}}$ such that

$$\left|\theta_{nom,i}(t) - \theta_{nom,i}^{(l)}\right| \leq \vartheta_{s,i} \tag{21}$$

where

$$\vartheta_{s,i} \leq \vartheta_{out} - k_{zi}\gamma \tag{22}$$

for $s = 1, \ldots, s$. This condition ensures that the reference trajectory has a sufficient number of design points so that the nominal reference trajectory is contained within $\cup_i \Theta_i$. In other words, this is an assumption that the reference trajectory has been sampled "adequately" and will be called the *adequate sampling assumption.*

4 Performance of Switched LPV Systems

At this point, an LPV model for the plant dynamics has been introduced. The assumption placed on this model was the existence of a known, nominal parameter trajectory representing ideal performance and/or scheduling objectives for the plant. Linear, state-feedback control agents are designed for LTI plant models obtained by fixing the plant parameter at design points taken from the nominal parameter trajectory. Associated with each of the control agents is a switching set "centered" at the associated design point; switches between control agents take place when the parameter trajectory evolves out of the switching set

associated with the control agent currently in feedback with the system. The new control agent is selected according to the nearest neighbor switching rule.

The following proposition is called the LPV switching lemma. By establishing sufficient conditions for bounded amplitude performance when the dynamics of a system switch between two LPV realizations, the LPV switching lemma provides the basis for the main results of the paper.

Proposition 4.1 (LPV Switching Lemma) *Consider any finite constants $r \in (0,1]$ and $\gamma > 0$, compact sets $\Theta_1, \Theta_2 \subset \tilde{\Theta}$, continuous matrix mappings $A_i : \mathbb{R}^s \to \mathbb{R}^{n\times n}$ and $B_i : \mathbb{R}^s \to \mathbb{R}^{n\times n_w}$ for $i = 1,2$ and constant matrices $C_i \in \mathbb{R}^{n_z\times n}$ for $i = 1,2$. Let $\mathcal{C} = \{\Theta_1, \Theta_2\}$.*

Suppose there exist constants $\alpha > 0$, $\beta > 0$, and $\rho > 0$ and positive definite matrices P_1 and P_2 such that

$$rP_2 \leq P_1 \tag{23}$$

$$\gamma^2 P_1 \geq C_1'C_1 \tag{24}$$

$$\gamma^2 P_2 \geq C_2'C_2 \tag{25}$$

$$P_1 \in \text{FeasRic}\left(A_1(\theta), B_1(\theta), 2\beta + \frac{\alpha}{r}, \alpha\right) \ \forall \theta \in \Theta_1 \tag{26}$$

$$P_2 \in \text{FeasRic}\left(A_2(\theta), B_2(\theta), \rho, \rho\right) \ \forall \theta \in \Theta_2 \tag{27}$$

Let w, x, and z be the input, state, and output, respectively, of the dynamical system

$$\begin{bmatrix} \dot{x}(t) \\ z(t) \end{bmatrix} = \begin{cases} \begin{bmatrix} A_1(\theta(t)) & B_1(\theta(t)) \\ C_1 & 0 \end{bmatrix} \begin{bmatrix} x(t) \\ w(t) \end{bmatrix}, & t \in [0, t_s) \\ \begin{bmatrix} A_2(\theta(t)) & B_2(\theta(t)) \\ C_2 & 0 \end{bmatrix} \begin{bmatrix} x(t) \\ w(t) \end{bmatrix}, & t \in (t_s, T] \end{cases} \tag{28}$$

where $t_s > 0$, $T \in (t_s, \infty)$. If $x'(0)P_1x(0) \leq 1$ and $w \in \mathcal{BL}_\infty^{n_w}$, then for any switching time satisfying

$$t_s > t_d := -\frac{1}{2\beta}\log r \tag{29}$$

with parameter trajectory $\theta(t) \in \mathcal{F}_\mathcal{C}$ satisfying $\theta(0) \in \Theta_1$ and $\theta(t_s^+) \in \Theta_2$,

$$\|z\|_{\infty,[0,T]} \leq \gamma.$$

The LPV switching lemma states three sufficient conditions for bounded amplitude performance of switch LPV error systems.

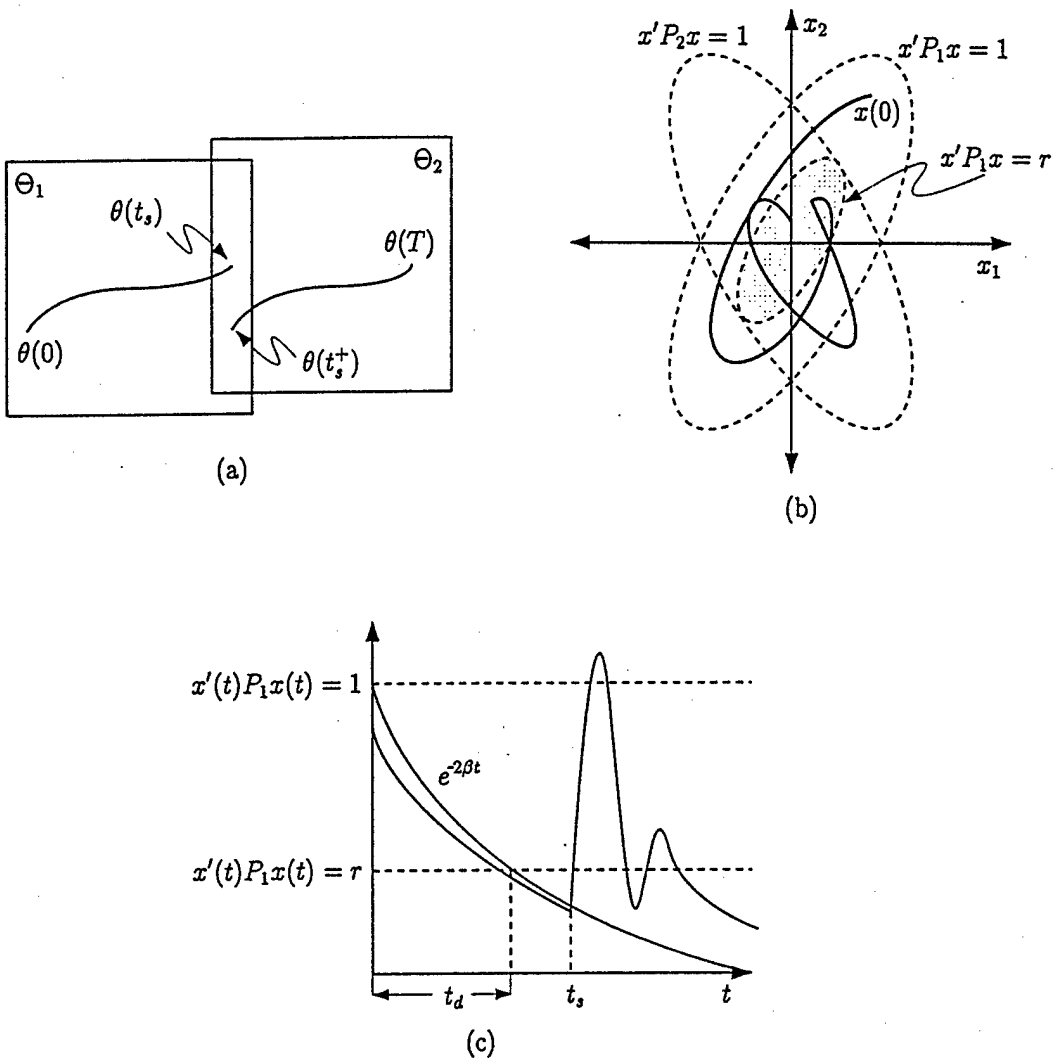


Figure 2: Illustration of LPV Switching Lemma

1. The first condition is that the initial error state, $x(0)$, lie in the ellipsoid $\{x|x'P_1x \leq 1\}$, as depicted in figure 2(b). From theorem 1 of [5], equations 24 and 26 are sufficient to guarantee that this ellipsoid is invariant for times prior to t_s and that any point $\tilde{x} \in \{x|x'P_1x \leq 1\}$ will also satisfy $\|C_1\tilde{x}\| \leq \gamma$. Thus, the first condition guarantees performance on the interval $[0, t_s]$ and is sufficient to ensure performance over the interval $[0, T]$ if no switch were to occur.

2. The second condition is that the parameter trajectories must evolve over the switching sets Θ_1 and Θ_2 so that $\theta(t) \in \mathcal{F}_C$ with the added restriction that $\theta(0) \in \Theta_1$ and $\theta(t_s^+) \in \Theta_2$. This added restriction guarantees that while the system dynamics correspond to (A_i, B_i, C_i), $i = 1, 2$, the parameter trajectory lies in the set Θ_i; this condition ensures that the Riccati inequalities of the lemma are valid for

the system. Figure 2(a) depicts a parameter trajectory $\theta(t)$ which satisfies $\theta(t) \in \Theta_1$, $t \in [0, t_s]$ and $\theta(t) \in \Theta_2$, $t \in [t_s, T]$ with a discontinuity at time t_s. The condition of the lemma states that immediately after the discontinuity, the parameter trajectory must satisfy $\theta(t_s^+) \in \Theta_2$ as shown.

3. The final condition is that the switching time must satisfy a *dwell-time* requirement, $t_s \geq t_d$. This guarantees that the state error has had sufficient time to decay so that any transient associated with the switch will not violate performance constraints. Figure 2(b) depicts this condition in terms of invariant ellipsoids. At the time of the switch, the invariant ellipsoid associated with the system dynamics switches to $\{x | x' P_2 x \leq 1\}$. The dwell-time condition guarantees that at time t_s, the state $x(t_s)$ lies on the interior of the new invariant by ensuring that the state has had sufficient time to decay to the shaded region in the figure which is an ellipsoid $\{x | x' P_1 x \leq r\} \subset \{x | x' P_2 x \leq 1\}$. This decay time is characterized by the constant β which parameterizes a bounding exponential, as shown in figure 2(c). The dwell-time is computed by determining when this bounding exponential satisfies $e^{-2\beta t} = r$.

Remarks:

- The LPV switching lemma can be generalized to systems with θ-dependent output matrices e.g., $C(\theta)$ and $D(\theta)$, in a straightforward manner.

- The initial condition and dwell-time constraints of the lemma are directly related to the concept of uniform ultimate boundedness ([14]). In particular, the conditions of the lemma are special cases of lemma 5.2 in [14], p213. The LPV switching lemma possesses the parameter variation condition which is key to the switching behavior considered in this paper. The parameter variation condition may be seen as enforcing the perturbation bound of the results in [14].

Of these conditions, only the last two require further discussion. Note that because of the integrator in the controller structure, the control input to the plant is continuous. Since the state is also continuous, no discontinuities in the parameter trajectory will occur. Thus, under the nearest neighbor switching rule, the adequate sampling assumption ensures that the second condition of the lemma is satisfied for the switched systems considered in this paper. The only remaining condition which must be verified for the switched system is the dwell-time constraint.

5 Switched System Performance Analysis

In this section, results are presented which show that if the parameter variations are "nice" (i.e. if the constants k_{zi} in (7) are small enough), then $|\theta(t) - \theta_{nom}(t)|$ is bounded by a known quantity. The result is important because, under the assumption of adequate sampling of $\theta_{nom}(t)$, it allows one to bound possible times of control agent switches using the nominal parameter trajectory. This result is stated in lemma 5.1 and corollary 5.1. Proposition 5.1 provides bounds on the switching times of the system using the parameter variation bounds. By comparing these bounds with the dwell-time requirements of the LPV switching lemma, a systematic approach for the extraction of a timed-automaton can be obtained (section 6).

5.1 Parameter Deviations

The following result bounds the deviation between the nominal parameter trajectory and actual parameter trajectory. This lemma is useful in estimating switching times.

Lemma 5.1 *Given performance level $\gamma > 0$ and the modified LPV system of (15-16), suppose that the control input to the system is given by $u(t) = K\hat{x}(t)$ where K is a constant gain matrix. Let $\bar{\theta}$ be a point on the nominal parameter trajectory and let Θ be any compact subset of $\bar{\Theta}$ containing $\bar{\theta}$. Suppose that there exists a positive definite matrix P and constants $\alpha > 0$, $\beta \geq 0$, and $0 < r \leq 1$ such that*

$$P \geq \frac{1}{\gamma^2} C'C \tag{30}$$

$$P \in \text{FeasRic}\left(A(\theta) + B_u K, B_w(\theta), 2\beta + \frac{\alpha}{r}, \alpha\right) \ \forall \theta \in \Theta \tag{31}$$

For any $T > 0$, if $\hat{x}'(0)P\hat{x}(0) \leq 1$ and $w \in B\mathcal{L}_\infty^{n_w}[0,T]$, then any parameter trajectory $\theta \in \mathcal{F}_\Theta[0,T]$ must also satisfy

$$|\theta_i(t) - \theta_{nom,i}(t)| \leq k_{zi}\gamma \max\{\sqrt{r}, e^{-\beta t}\} \quad \text{for } i = 1, 2, \ldots, s \text{ and for all } t \in [0,T] \tag{32}$$

The implication of lemma 5.1 is that if the controllers are appropriately designed (so that (30) and (31) are satisfied), $\|z(t)\|$ will be bounded and therefore the parameter trajectory must remain bounded to the nominal parameter trajectory from (7). This reinforces the intuitive notion that parameter deviation from the nominal parameter trajectory represents modeling error. The bound is illustrated in figure 3 where the parameter set Θ is shown as a subset of a larger set $\bar{\Theta}$. According to the bound of the lemma, at any time

t, $\theta(t)$ must lie in a box centered at $\theta_{nom}(t)$ as defined by equation 32. As time increases, the box shrinks according to equation 32, tightening the bound.

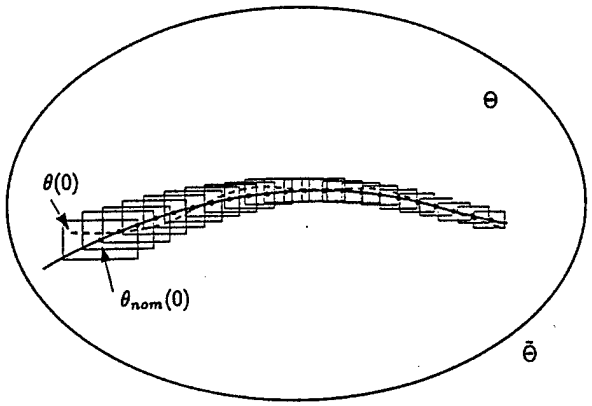


Figure 3: Illustration of parameter trajectory bounds.

Lemma 5.1 is important because it implies that if the bound on the parameter trajectory is small enough, then the nominal parameter trajectory represents a reasonable approximation to the true parameter trajectory. (Choosing the parameters r and β required to evaluate the bounds in discussed in appendix B.) This further implies that the nominal parameter trajectory may be used to bound switching times in the multiple agent control system. Corollary 5.1 indicates how the switching times may be estimated for the class of switching sets assumed in this paper.

Corollary 5.1 *Suppose that the conditions of lemma 5.1 are satisfied with adequate sampling and let* $T \in [0,\infty)$. *Let* $\Theta \subset \tilde{\Theta}$ *be a switching set. If* $\hat{x}'(0)P\hat{x}(0) \leq 1$, $w \in \mathcal{BL}_\infty^{n_w}[0,T]$ *and a parameter trajectory* $\theta \in \mathcal{F}_\Theta[0,T]$ *satisfies*

$$\max_{1\leq i\leq s} |\theta_i(T) - \bar{\theta}_i| = \vartheta_{out} \tag{33}$$

at time T, *then the nominal parameter trajectory at time* T *satisfies*

$$\vartheta_{out} - k_{zi}\gamma \max\{\sqrt{r}, e^{-\beta T}\} \leq |\theta_{nom,i}(T) - \bar{\theta}_i| \leq \vartheta_{out} + k_{zi}\gamma \max\{\sqrt{r}, e^{-\beta T}\} \tag{34}$$

for $i = 1, 2, \ldots, s$.

Corollary 5.1 is useful because it implies that events in the parameter space (e.g. the parameter trajectory crossing the boundary of a switching set) can be predicted with the nominal parameter trajectory. The time at which the parameter trajectory may intersect the boundary of Θ may be approximated by the times at which the nominal parameter trajectory evolves over points near the boundary of Θ. This is illustrated in

figure 4; when the conditions of the corollary are satisfied and if the true parameter trajectory intersects the boundary of Θ at time T, then the nominal parameter trajectory at time T must lie in the shaded region representing the bounds of (34).

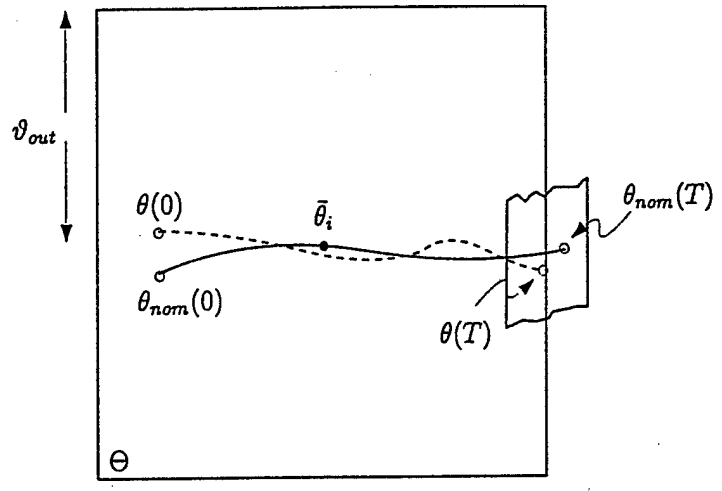


Figure 4: Estimation of switching times using $\theta_{nom}(t)$. The shaded region corresponds to the bounds given in equation 34.

5.2 Switching Time Estimation

When combined with a specific switching rule such as the nearest neighbor rule, the nominal parameter trajectory can be used to estimate switching times and the results of possible switches. This is apparent from corollary 5.1. The following result provides these estimates.

Proposition 5.1 *Given a performance level $\gamma > 0$ and the modified LPV system of (15-16), let $\mathcal{K}$ be a set of control agents which forms a control policy with the nearest neighbor switching rule under adequate sampling. Suppose that at time t_0, the control input to the system is given by $u(t_0) = K^{(l)}\hat{x}(t_0)$ where $K^{(l)} \in \mathcal{K}$ is a constant gain matrix. Let Θ_l be a switching set and suppose there exist positive definite matrix P and constant $\alpha > 0$, $\beta \geq 0$, and $0 < r \leq 1$ such that*

$$P \geq \frac{1}{\gamma^2} C'C \tag{35}$$

$$P \in \text{FeasRic}\left(A(\theta) + B_u K^{(l)}, B_w(\theta), 2\beta + \frac{\alpha}{r}, \alpha\right) \quad \forall \theta \in \Theta_l \tag{36}$$

Define the sets

$$\hat{\Theta}_m := \left\{ \theta \;\middle|\; \theta \in \tilde{\Theta}, \|\theta - \theta_{nom}^{(m)}\| \leq \|\theta_i - \theta_{nom}^{(q)}\|, m, q \in \mathcal{I}_{\mathcal{K}}, q \neq m \right\}$$

and

$$\mathcal{T}^{(l,m)} := \left\{ t \;\middle|\; \begin{array}{l} |\theta_{nom,i}(t) - \hat{\theta}_i| \le k_{max,i}(t) \quad \textit{for } i = 1,\ldots,s \textit{ and some } \hat{\theta} \in \hat{\Theta}_m, \textit{ and} \\ \vartheta_{out} - k_{max,i}(t) \le |\theta_{nom,i}(t) - \theta^{(l)}_{nom,i}| \le \vartheta_{out} + k_{max,i}(t), \quad \textit{for some } 1 \le i \le s \end{array} \right\}$$

where

$$k_{max,i}(t) := k_{zi}\gamma \max\{\sqrt{r}, e^{-\beta(t-t_0)}\} \tag{37}$$

If $\hat{x}'(t_0)P\hat{x}(t_0) \le 1$, $w \in \mathcal{BL}_\infty$, *and a parameter trajectory* θ *is generated by the nearest neighbor switching rule under adequate sampling, then the switch time,* t_s, *between the* l*th and* m*th systems satisfies* $t_s \in \mathcal{T}^{(l,m)}$.

There are two primary components to the construction of a set $\mathcal{T}^{(l,m)}$:

Switching Destinations The set $\hat{\Theta}_m$ represents the set of all parameter vectors $\theta \in \tilde{\Theta}$ which satisfy

$$m = \arg\min_{j \in \mathcal{I}_\mathcal{K}} \left\| \theta - \theta^{(j)}_{nom} \right\|;$$

if the parameter trajectory at time t_s lies in $\hat{\Theta}_m$, a switch to control agent m will take place according to the nearest neighbor switching rule. The sets $\hat{\Theta}_m$ may be represented by a set of affine inequality constraints on the parameter vector θ, as depicted by the light-shaded regions of figure 5. Note that these constraints may be computed in an off-line fashion with knowledge of the nominal parameter trajectory and the design points. (See [6] for details.)

Since the true parameter trajectory can only be estimated by the nominal parameter trajectory (as a consequence of lemma 5.1), the times for which $\theta(t) \in \hat{\Theta}_m$ can only be estimated. Let the set

$$\bar{\Theta}_t := \{\theta \mid |\theta_{nom,i}(t) - \theta| \le k_{max,i}(t), \text{ for all } i = 1,\ldots,s\}$$

For each time t, the set $\bar{\Theta}_t$ is a hyper-rectangle centered at $\theta_{nom}(t)$ which contains the true parameter trajectory $\theta(t)$, as depicted in figure 5. The first requirement for a time t to belong to $\mathcal{T}^{(l,m)}$ is for the intersection $\bar{\Theta}_t \cap \hat{\Theta}_m$ to be nonempty, which is a relatively simple convex feasibility problem. In figure 5, this requirement is satisfied at times t_1, t_2 and t_3; the requirement is not satisfied at time t_0.

Switching Times As a consequence of corollary 5.1, the times for which the nominal parameter trajectory satisfies

$$\vartheta_{out} - k_{max,i}(t) \le |\theta_{nom,i}(t_s) - \theta^{(l)}_{nom,i}| \le \vartheta_{out} + k_{max,i}(t)$$

for some $i = 1, \ldots, s$, bound the times for which the true parameter trajectory may intersect the boundary of the parameter set Θ_i. This condition represents an added restriction for a time t to belong to $\mathcal{T}^{(l,m)}$. Note that this requirement may also be expressed with a set of affine inequality constraints on the parameter vector θ which may be computed in an off-line fashion. An illustration of these sets is indicated by the dark shaded regions in figure 5. In the figure, the requirement is satisfied by time t_3; times t_0, t_1 and t_2 do not satisfy the switching requirement.

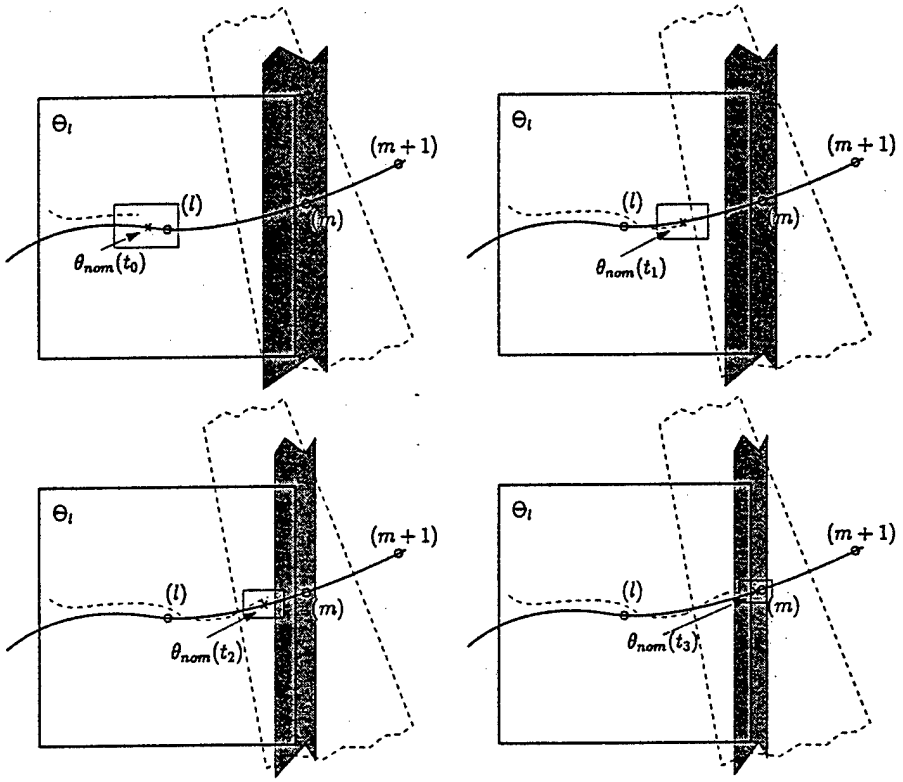


Figure 5: Illustration of $\mathcal{T}^{(l,m)}$. Light regions represent $\hat{\Theta}_m$, dark regions represent switching bounds from corollary 5.1, and small box, $\bar{\Theta}_t$, represents bounds on parameter variation from lemma 5.1. Time $t \in \mathcal{T}^{(l,m)}$ if and only if $\theta_{nom}(t)$ lies in the dark shaded region *and* $\bar{\Theta}_t \cap \hat{\Theta}_m$ is nonempty. Here, $t_0, t_1, t_2 \notin \mathcal{T}^{(l,m)}$ and $t_3 \in \mathcal{T}^{(l,m)}$.

Given the preceding descriptions, the set $\mathcal{T}^{(l,m)}$ is constructed by finding all times for which the nominal parameter trajectory lies near the switching surface (i.e., lies in the dark-shaded region of figure 5) *and* for which the corresponding set $\bar{\Theta}_t$ has a nonempty intersection with $\hat{\Theta}_m$ (i.e., times for which the small rectangles of figure 5 intersect the light-shaded region). Constructing a set $\mathcal{T}^{(l,m)}$ amounts, essentially, to conducting a line search over the nominal parameter trajectory, evaluating a set of convex constraints at

each point. Note that the computation is performed off-line; time and computational resources are not a significant issue.

It is apparent from the construction of the switching-time sets of proposition 5.1 and from the nearest neighbor switching rule that certain switches will never take place due to the geometry of the parameter space. In fact, given that control agent $K^{(l)}$ is currently in feedback with the system, under the assumption of adequate sampling, then a switch to control agent $K^{(m)}$ should take place if and only if $m \in \mathcal{I}_l$ where

$$\mathcal{I}_l := \left\{ m \,\middle|\, m \neq l, \Theta_l \cap \hat{\Theta}_m \neq \emptyset \right\}. \tag{38}$$

Such a switch from agent $K^{(l)}$ to agent $K^{(m)}$ will be called an *admissible switch.*

5.3 Stability and Performance Results

In this section, properties of the continuous-state system, namely bounded-amplitude performance results for LPV systems, are used to derive conditions for switched systems to satisfy performance constraints. These conditions amount to verifying that transitions between states of the automaton do not occur too quickly. More specifically, the transitions must satisfy the *dwell-time* constraint of the LPV switching lemma. The following result states this condition.

Proposition 5.2 *Given a performance level $\gamma > 0$ and modified LPV system of (15-16), let $\mathcal{K}$ be a set of control agents which form a control policy with the nearest neighbor switching rule under the assumption of adequate sampling.*

Suppose that for each $l \in \mathcal{I}_{\mathcal{K}}$, there exists a positive definite matrix $P^{(l)}$ and constants $\alpha_1^{(l)} \geq \alpha_2^{(l)} > 0$ such that

$$P^{(l)} \geq \frac{1}{\gamma^2}\tilde{C}'\tilde{C} \tag{39}$$

$$P^{(l)} \in \text{FeasRic}\left(\tilde{A}(\theta) + \tilde{B}_v K^{(l)}, \tilde{B}_w(\theta), \alpha_1^{(l)}, \alpha_2^{(l)}\right), \ \forall \theta \in \Theta_l \tag{40}$$

Denote the agent initially in the feedback loop by $K^{(0)}$. If all possible switches are admissible, $\hat{x}'(0)P^{(0)}\hat{x}(0) \leq 1$ and if, for all admissible switching sequences $k \to l \to m$,

$$\frac{r^{(l,m)}}{\alpha_2^{(l)} - \alpha_1^{(l)} r^{(l,m)}} \log r^{(l,m)} \leq \min \mathcal{T}^{(l,m)} - \max \mathcal{T}^{(k,l)} \tag{41}$$

where

$$r^{(l,m)} := \max\left\{ r \;\middle|\; rP^{(m)} \leq P^{(l)}, \frac{\alpha_2^{(l)}}{\alpha_1^{(l)}} < r \leq 1 \right\} \tag{42}$$

then for any $T \in [0, \infty)$,

$$\|z\|_{\infty,[0,T]} \leq \gamma \tag{43}$$

and for $T \to \infty$,

$$\|z\|_{\infty,[0,\infty)} \leq \gamma. \tag{44}$$

Proposition 5.2 is interpreted as follows. Denote

$$\mathcal{R}^{(l,m)} := \left\{ r \;\middle|\; rP^{(m)} \leq P^{(l)}, \frac{\alpha_2^{(l)}}{\alpha_1^{(l)}} < r \leq 1 \right\} \tag{45}$$

The set $\mathcal{R}^{(l,m)}$ represents the set of all possible constants r which yield a positive dwell-time, as defined in the LPV switching lemma, for a switch between control agents $K^{(l)}$ and agent $K^{(m)}$. Similarly, the quantity

$$\min \mathcal{T}^{(l,m)} - \max \mathcal{T}^{(k,l)}$$

represents the minimum time that can elapse between a switches from $k \to l$ and $l \to m$. Equation (41) therefore represents the dwell-time constraint of the LPV switching lemma verified using the switching-time estimates obtained from proposition 5.1. Thus, under the adequate sampling assumption and assuming that the initial state error is small enough, it is sufficient to check that admissible switching sequences satisfy the dwell-time constraint to ensure bounded amplitude performance.

By applying the results of proposition 5.1 to obtain the switching sets $\mathcal{T}^{(l,m)}$, proposition 5.2 may be applied to analyze bounded amplitude performance in the finite-time and periodic scheduling problems stated in section 3.2. (Note that proposition 5.2 applies to non-periodic $\theta_{nom}(t)$ defined for $t \in [0, \infty)$ as well.)

6 Applications to Hybrid Systems

The switched agent control system is a *hybrid system* because it generates a mixture of discrete event and continuous-valued signals. As noted in section 1, hybrid systems can be studied from two distinct viewpoints; as a supervised collection of real-time computer processes or as switched dynamical systems. The integration of these two viewpoints is required for the development of useful and efficient methodologies for hybrid system design. While performance analysis for a given LPV system may best be carried out in the continuous domain, integrating the controlled system with other systems may best be accomplished on the supervisory level. In this section, we show how the two viewpoints are related for the class of systems considered in this

paper through the bounded amplitude performance analysis described above. Specifically, we show how the switched system performance analysis of the preceding sections supplies information sufficient to construct a timed logical model, in the form of a timed automaton, of the switched system behavior.

Finite automata represent a powerful symbolic modeling tool for supervisory controlled systems. Modeling systems in such a way leads to effective procedures for automatically manipulating and analyzing system behavior at the supervisory level. However, such a model lacks the power to express real-time behaviors. Timed automata [2] arose out of the desire to extend this modeling ability to the verification of real-time systems.

In the remainder of this section, we show how the results of the switched system performance analysis conducted using a continuous-time control theoretic approach, may be used to construct a timed-automaton model of the controlled system. This *automaton extraction* provides part of the aforementioned link between the two approaches for analysis and design of hybrid systems. We begin with a brief description to timed-automata and we illustrate the automaton extraction with a simple example which also serves to illustrate the results presented in the previous sections. Finite-time and periodic scheduling objectives are considered separately.

6.1 Timed-Automata

A *finite automaton* is characterized by the ordered pair, $\mathcal{N} = (\mathcal{V}, \mathcal{A})$ where $\mathcal{V}$ is a finite set of $M_{\mathcal{V}}$ vertices and $\mathcal{A} \subseteq \mathcal{V} \times \mathcal{V}$ is a set of directed arcs between vertices. The automaton, $\mathcal{N}$, is marked by a function $\mu : \mathcal{V} \to \{0, 1\}$. The marking function, μ, is said to be valid if and only if there is at most one $p \in \mathcal{V}$ such that $\mu(p) = 1$. The vector $\bar{\mu} = [\mu(p_1), \ldots, \mu(p_{M_{\mathcal{V}}})]$ is used to represent the state of the automaton. A marked automaton is then represented by the ordered triple, $(\mathcal{V}, \mathcal{A}, \bar{\mu}_0)$, where $\bar{\mu}_0$ is the initial marking vector of the automaton.

The dynamic behavior of the automaton is generated by the firing of arcs. An arc $(p, q) \in \mathcal{A}$ is said to be enabled if $\mu(p) = 1$. An enabled arc is free to fire. Let $\bar{\mu}$ and $\bar{\mu}'$ be the marking vectors of the automaton before and after the firing of arc (q_0, q_1), respectively. The relationship between these marking vectors is given by

$$\bar{\mu}(p) = \begin{cases} 1 & \text{if } p = q_0 \\ 0 & \text{otherwise} \end{cases} \quad \text{and} \quad \bar{\mu}'(p) = \begin{cases} 1 & \text{if } p = q_1 \\ 0 & \text{otherwise} \end{cases}$$

where $\bar{\mu}(p)$ and $\bar{\mu}'(p)$ represent the pth elements of $\bar{\mu}$ and $\bar{\mu}'$, respectively.

A *timed automaton* arises by introducing a finite set of clocks, $\mathcal{X}$, and by introducing mappings which label the arcs and vertices of the automaton $\mathcal{N} = (\mathcal{V}, \mathcal{A})$ with equations representing constraints on the clock state. The ith clock will be characterized by the ordered triple $\mathcal{X}_i = (c_i, x_{i0}, \tau_{i0})$ where $x_{i0} \in \mathbb{R}^n$, $\tau_{i0} \in \mathbb{R}$ and $c_i \in \mathbb{R}^n$. The *local time* of the ith clock, $x_i(\tau)$, $(\tau \geq \tau_{i0})$ generated by clock $\mathcal{X}_i$ is the solution to the initial value problem

$$\dot{x}_i(t) = c_i; \;\; x_i(\tau_{i0}) = x_{i0} \tag{46}$$

The set of all local times and clock rates at time τ will be called the *clock state* and will be denoted

$$\bar{x}(\tau) = \{(x_i(\tau), c_i)\}_{i=1,2,\ldots,M_\mathcal{V}} \tag{47}$$

Let $\mathcal{P}$ be a set of formulae defined over the clock state, $\bar{x}(\tau)$. We say that the clock state $\bar{x}(\tau)$ satisfies a formula $p \in \mathcal{P}$ if the formula is true for the current state assignment at time τ. This is denoted as $\bar{x}(\tau) \models p$. A simple example of this is the comparison of a clock value to some threshold, e.g. $x_i(\tau) > \tau_c$. In the examples which follow, a local clock, t, must lie in a closed interval, e.g. $t \in [0.0508, 0.0519]$, or the clock is reset to zero, $t \leftarrow 0$. A *timed automaton* is formally defined by the tuple, $(\mathcal{N}, \mathcal{X}, \ell_f, \ell_r, \ell_v)$ where

- $\mathcal{N} = (\mathcal{V}, \mathcal{A}, \bar{\mu}_0)$ is a finite automaton with initial marking vector $\bar{\mu}_0$.
- $\ell_f : \mathcal{A} \to \mathcal{P}$ is the *firing condition.* For an arc $(p, q) \in \mathcal{A}$, $\bar{x} \models \ell_f(p, q)$ means that the arc (p, q) is free to fire provided that it is already enabled.
- $\ell_v : \mathcal{V} \to \mathcal{P}$ is the *vertex constraint.* If $\bar{x} \models \ell_v(v)$ for some $v \in \mathcal{V}$, then clock states are forced to satisfy an equality constraint while $\mu(v) = 1$.
- $\ell_r : \mathcal{A} \to \mathcal{P}$ is the *reset constraint.* For an arc $(p, q) \in \mathcal{A}$, this mapping represents an equality constraint which the clock state is reset to immediately after the firing of arc (p, q).

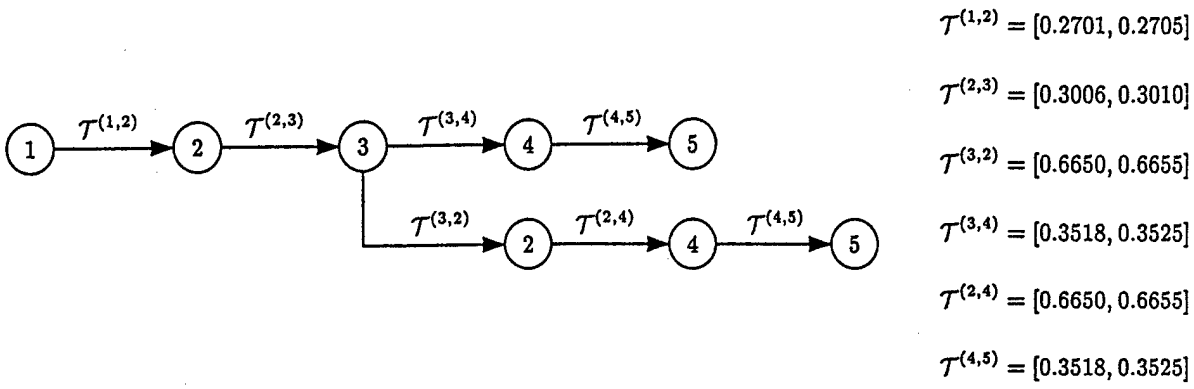
The mappings ℓ_f, ℓ_r and ℓ_v all represent constraints on the clock states which must be satisfied for transitions to occur. In the examples that follow, the vertex constraint is simple because the clocks do not change. In a more general setting, clock constraints can change as the system evolves. The clocks themselves can also be defined as the solution of a more general set of differential equations; the timed automaton is then referred to as a *hybrid automaton* [1]. In a model reference control problem, one may view the reference model as a clock. The vertex constraint then corresponds to changing the reference model.

The preceding definition of timed automata is essentially the same as that used in [2]. The description provided above, however, follows notational conventions found in the Petri net literature and appears to be

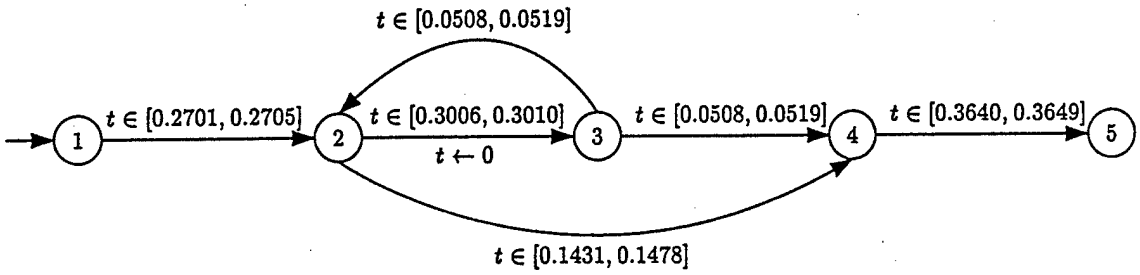
more closely related to the control theoretic approach described earlier. As noted, finite automata represent powerful symbolic modeling tools for supervisory controlled systems. They are useful because efficient computational algorithms exist to verify that a finite automaton satisfies behaviour specifications which can be posed as formulae in computational tree logic [11]. This verification procedure, known as symbolic model checking [16] has provided a powerful tool in the verification of VLSI digital circuits[9]. Timed [2] and hybrid automata [1] arose from a desire to extend symbolic model checking to the verification of real-time systems.

Now consider the application of proposition 5.2 to the multiple agent system. To apply the proposition, one needs to construct the switch-time sets, $\mathcal{T}^{(l,m)}$, and verify that the dwell-time constraints are satisfied (41). If these sets can be constructed, one may next construct a sequential model of the switching behavior with a tree structure. The nodes of the tree correspond to a control agent being switched into feedback with the plant. The nodes can be collected into levels. Each level, $l_s \in \{0, \ldots, N\}$, of the tree contains possible states of the system after l_s switches have taken place, i.e. each level contains the indices of control agents in the loop after l_s switches have taken place. In turn, a state l in level l_s is connected to a state m in level l_s+1 by an arc labeled with the time interval over which the switch could possibly take place, $\mathcal{T}^{(l,m)}$. Because this model is constructed while ensuring that the bounded amplitude performance constraint is satisfied, we will call this model a *performance validation tree.* A simple illustration of a performance validation tree is shown in figure 6(a). (This example was presented in [15].) The tree of figure 6(a) is initialized with a node labeled 1, indicating that agent $K^{(1)}$ is initially in the feedback loop. A switch from controller $K^{(1)}$ to $K^{(2)}$ will take place during the time interval $t \in \mathcal{T}^{(1,2)}$; this is indicated on the tree by the branch from node 1 to node 2 labeled with $\mathcal{T}^{(1,2)}$. Where more than one branch leaves a node, a nondeterministic switch is indicated. In the figure, during the time interval $t \in [0.3518, 0.3525]$, a switch from $K^{(3)}$ to either $K^{(2)}$ or $K^{(4)}$ is possible. The remaining portions of the tree are interpreted in a similar fashion.

The performance validation tree is now used to construct a timed-automaton model for the switched agent system. Suppose that a control agent, $K^{(l)}$, in feedback with the plant is seen as a *state* of the controlled system. Then, assuming that a finite number of switches occurs over a finite interval, $[0, T]$, a finite sequence of states will be reached by the controlled plant during $[0, T]$. The performance validation tree described above represents all possible finite sequences of states which can be assumed by the closed-loop system over $[0, T]$. In other words, the performance validation tree represents all possible trajectories of a *timed-automaton* model of the multiple agent controlled system. A graphical representation of the timed-automaton corresponding to the performance validation tree of figure 6(a) is the timed-transition table shown



(a)



(b)

Figure 6: Illustration of sequential models of switching behavior: (a) Performance validation tree (b) Timed transition table

in 6(b). The finite automaton $\mathcal{N}$ is given by the states $1-5$ and the indicated directed arcs connecting the states. The automaton is initialized with a marking vector $\vec{\mu}_0 = [1,0,0,0,0]$. The local clock, t, is reset on initialization and on the firing of arc $(2,3)$. The firing conditions are derived from the sets $\mathcal{T}^{(l,m)}$ with the reset condition taken into account. For example, from figure 6(a), the local clock reset occurs at some point in the interval $\mathcal{T}^{(2,3)}$. If arc $(3,2)$ is to fire, it must occur in the interval $\mathcal{T}^{(3,2)}$. With respect to the local clock, this time interval becomes

$$\left[\min \mathcal{T}^{(3,2)} - \max \mathcal{T}^{(2,3)}, \max \mathcal{T}^{(3,2)} - \min \mathcal{T}^{(2,3)}\right]$$

which is indicated in figure 6(b).

6.2 Finite-time Scheduling

We now turn to numerical examples illustrating the results presented above. The plant chosen for the purposes of illustration is a second-order nonlinear system representing a typical chemical process control problem given by

$$\dot{x}_{p1} = -x_{p1} + v_1 \tag{48}$$

$$\dot{x}_{p2} = -x_{p2} + (1 + x_{p1}^2)v_2 \tag{49}$$

The finite-time scheduling objective considered here is to move the state of the plant from points near $x_p = (2.5, 2)$ to points near $x_p = (1, 3)$ in one second ($T = 1$) according to the reference model

$$\dot{x}_{m1} = -1.5 \tag{50}$$

$$\dot{x}_{m2} = 1 \tag{51}$$

By defining $x := x_m - x_p$, a quasi-LPV description of the error system may be obtained from

$$\dot{x} = A(\theta)x + B_w(\theta)w + B_v(\theta)v \tag{52}$$

where

$$A(\theta) = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix}, \; B_w(\theta) = \begin{bmatrix} \frac{3}{4}\theta_2 + \frac{1}{4} \\ \frac{1}{2}\theta_3 + \frac{7}{2} \end{bmatrix} \text{ and } B_v(\theta) = \begin{bmatrix} -1 & 0 \\ 0 & -\frac{21}{8}\theta_1 - \frac{37}{8} \end{bmatrix} \tag{53}$$

The parameter mapping was chosen as

$$\begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \end{bmatrix} = \begin{bmatrix} \frac{8}{21}x_{p1}^2 - \frac{29}{21} \\ \frac{4}{3}x_{m1} - \frac{7}{3} \\ 2x_{m2} - 5 \end{bmatrix} \tag{54}$$

so that the nominal parameters would all vary between -1 and 1. Here, $w = 1$ is introduced as a fictitious disturbance so that the nonlinearities grouped in the $B_w(\theta)$ term are treated as a bounded disturbance. The performance constraint is given by $\|z\|_{\infty,[0,T]} \leq \gamma$ where $z = C_1 x + D_{12} v$ with $C_1 = I$ and $D_{12} = 0.01I$. Performance is considered for various levels of γ and various numbers of agents.

Control agents take the form

$$\dot{v} = u \tag{55}$$

$$u = K_1^{(l)}x + K_2^{(l)}(v + v^{(l)}) \tag{56}$$

where the index l indicates the design point and $v^{(l)}$ is a constant bias term internal to each agent. The bias term for each design point $\theta^{(l)}$ is chosen as

$$v^{(l)} = -B_v(\theta^{(l)})^{-1}B_w(\theta^{(l)}) \tag{57}$$

Note that the inverse exists in this case for all nominal parameter values.

Control gains were synthesized for the biased systems using the techniques presented in [5] combined with LMI pole placement constraints ([10]). A MATLAB program was written to implement the conditions associated with propositions 5.1 and 5.2. First, the switch-time sets $\mathcal{T}^{(l,m)}$ were computed according to the conditions of proposition 5.1. The nominal parameter trajectory was searched to determine possible switching times and the resulting switches. The results of the search were used to form a performance validation tree for a fixed performance level γ and switching parameter ϑ_{out}. Figure 7 depicts one such tree for a multiple agent design with switching parameter $\vartheta_{out} = 0.4$. For this design, 7 control agents were required to satisfy the adequate sampling assumption. The performance validation tree shown in the figure represents a performance level of $\gamma = 0.068$. The performance validation tree is initialized with control

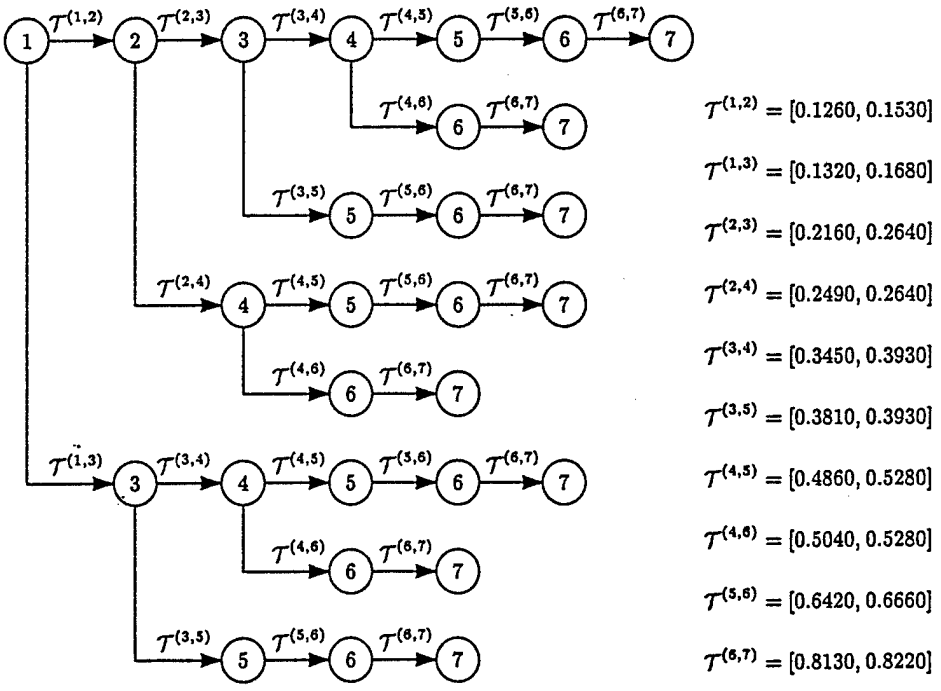


Figure 7: Performance validation tree for finite-time example: $\vartheta_{out} = 0.4$, $\gamma = 0.068$

agent $K^{(1)}$ in feedback with the plant. A search of $\theta_{nom}(t)$ indicated a possible switch to agent $K^{(2)}$ for $t \in \mathcal{T}^{(1,2)}$ or to agent $K^{(3)}$ for $t \in \mathcal{T}^{(1,3)}$. This result is indicated by the two branches leaving node 1 and

labeled with $\mathcal{T}^{(1,2)}$ and $\mathcal{T}^{(1,3)}$. The second level of the tree is constructed by first assuming that agent $K^{(2)}$ is switched into feedback at time $\max \mathcal{T}^{(1,2)}$; $\theta_{nom}(t)$ is then searched, resulting in nonempty sets $\mathcal{T}^{(2,3)}$ and $\mathcal{T}^{(2,4)}$ indicating possible switches to agents $K^{(3)}$ or $K^{(4)}$ from agent $K^{(2)}$. The process is repeated until the time interval is exhausted with no further switches.

The performance validation tree of figure 7 was then used to construct an automaton model which is represented by the timed transition table shown in figure 8. For the underlying finite automaton, $\mathcal{V}$ consists

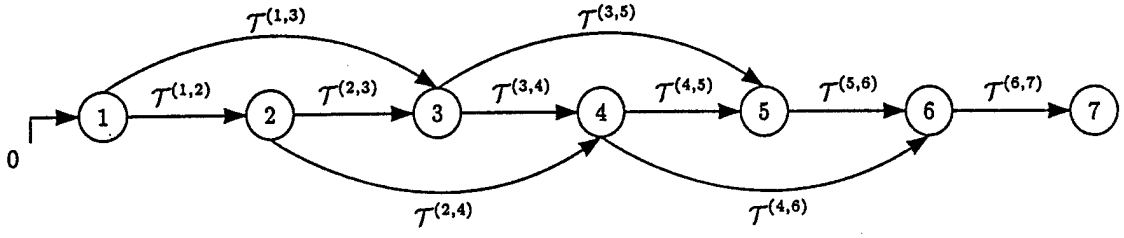


Figure 8: Timed transition table for finite-time example: $\vartheta_{out} = 0.4$, $\gamma = 0.068$.

of seven states with connecting directed arcs, $\mathcal{A}$, as indicated in the figure. The initial marking vector satisfies $\mu(1) = 1$, i.e. the automaton is initialized in state 1. There are no reset conditions on the single local clock t. Firing constraints are indicated in the figure, e.g. $t \in \mathcal{T}^{(3,4)}$.

The timed-automaton represented in figure 8 represents an abstraction of of the multiple agent controlled system which can be analyzed on the supervisory level. If the switched system described here is one of many similar subsystems, this timed-logical model would be useful in the verification of, for example, desired synchronous behavior among the subsystems.

It is useful at this time to remark on the performance improvement observed with increasing the number of agents (decreasing ϑ_{out}). The results of several different designs produce the plot in figure 9 which shows the minimum level γ_{ver} for which performance of the switched system could be guaranteed with proposition 5.2 versus the number of agents used for control. The error bars in the plot indicate the precision of the result; performance conditions were satisfied to the top of the error bar, the conditions failed at the bottom of the error bar. Figure 9 clearly demonstrates monotonic improvement in performance with increasing number of agents. This trend was also observed in simulations of the corresponding closed loop systems.

One reason for the observed improvement can be seen by comparing the control input to the plant generated by the switched agent controller with the model-matching control

$$v_{ref1} = x_{p1} - 1.5 \tag{58}$$

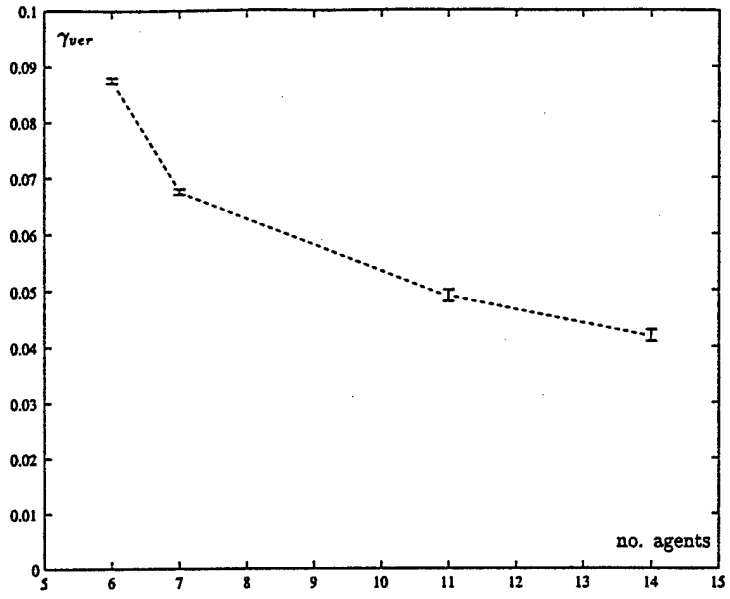


Figure 9: Verified performance versus number of agents.

$$v_{ref2} = \frac{x_{p2}+1}{1+x_{p1}^2} \tag{59}$$

evaluated for the system with zero initial state error. Figure 10 plots the average difference between the actual and reference control taken over the time interval $[0,1]$. The increasing number of agents allows new agents to be switched into feedback before the state error, hence the control effort, deviates from the reference.

6.3 Periodic Scheduling

The periodic scheduling objective considered here is to cycle the state of the plant described by (48-49) between points near $x_p = (0.5, 0.5)$ and points near $x_p = (1, 0.5)$ and back with a one second ($T = 1$) period according to the reference model

$$\dot{x}_{m1} = \frac{\pi}{2}\sin 2\pi t; \;\; x_{m1}(0) = 0.5; \tag{60}$$

$$\dot{x}_{m2} = \frac{\pi}{4}\cos 2\pi t; \;\; x_{m2}(0) = 0.5; \tag{61}$$

As before, by defining $x := x_m - x_p$, a quasi-LPV description of the error system may be obtained from

$$\dot{x} = A(\theta)x + B_w(\theta)w + B_v(\theta)v \tag{62}$$

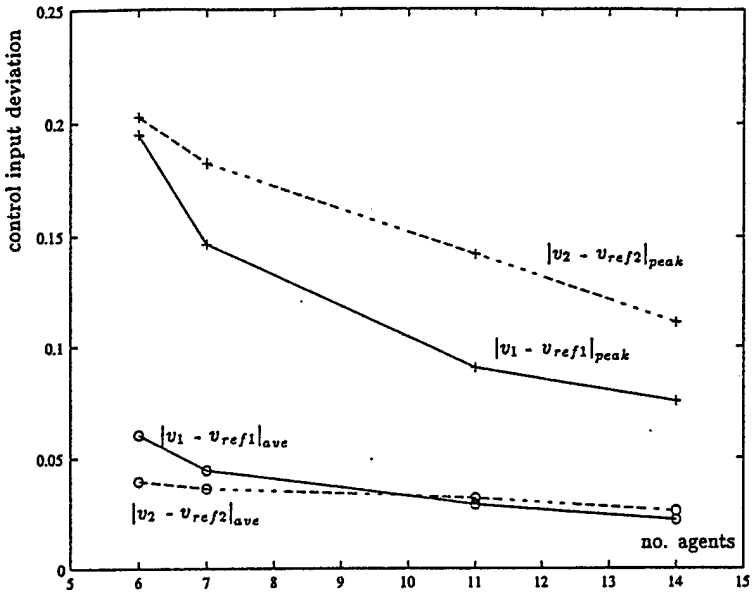


Figure 10: Deviation in control input, v, from model-matching control, v_{ref}, versus number of agents.

where

$$A(\theta) = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix}, \quad B_w(\theta) = \begin{bmatrix} \frac{7}{4}\theta_2 + \frac{3}{4} \\ \frac{9}{10}\theta_3 + \frac{1}{2} \end{bmatrix} \text{ and } B_v(\theta) = \begin{bmatrix} -1 & 0 \\ 0 & -\frac{3}{8}\theta_1 - \frac{13}{8} \end{bmatrix} \tag{63}$$

The parameter mapping was chosen as

$$\begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \end{bmatrix} = \begin{bmatrix} \frac{8}{3}x_{p1}^2 - \frac{5}{3} \\ \frac{4}{7}x_{m1} + \frac{\pi}{7} - \frac{8\pi-3}{7} \\ -\frac{10\pi}{9}x_{m1} + \frac{10}{9}x_{m2} + \frac{15\pi-10}{18} \end{bmatrix} \tag{64}$$

so that the nominal parameters would all vary between -1 and 1. Here, we have used the fact that the solution to the differential equations for the reference model yields

$$\cos 2\pi t = -4x_{m1} + 3 \qquad \sin 2\pi t = 8x_{m2} - 4$$

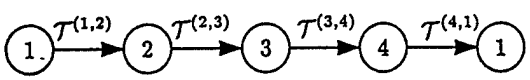
As before, $w = 1$ is introduced as a fictitious disturbance so that the nonlinearities grouped in the $B_w(\theta)$ term are treated as a bounded disturbance. The performance constraint is given by $\|z\|_{\infty,[0,T]} \leq \gamma$ where $z = C_1 x + D_{12} v$ with $C_1 = I$ and $D_{12} = 0.01I$.

The control agents used for the periodic scheduling possessed a structure identical to that used for the finite-time scheduling. The control synthesis procedure was identical as well.

Because the reference trajectory is periodic and the nominal parameter trajectory evolves over an infinite time horizon, one cannot apply the results of proposition 5.1 to compute the sets $\mathcal{T}^{(l,m)}$ in the same manner

as for the finite-time case. This difficulty arises because (37) can no longer be evaluated for all times $t \geq t_0$. However, one can use a more conservative bound by setting $k_{max,i}(t) = k_{zi}\gamma$ in (37). Any predicted switch satisfying the dwell-time conditions in this more conservative approach must also satisfy the dwell-time constraints if the less conservative bound had been used.

Using this more conservative approach, we need only evaluate admissible switches for single period trajectories, initialized with agents consistent with the established switching rules. The switching time estimates represent absolute bounds on when a switch might occur, regardless of the time that the agent was switched into the loop. As an example, for $\vartheta_{out} = 0.8$, four agents were sufficient to adequately sample the nominal parameter trajectory which was searched to determine possible switching times over a single period of the trajectory using the modified bounds described above. The resulting performance validation tree describing the possible switching behavior for a single period is shown in figure 11. The tree indicates



$$\mathcal{T}^{(1,2)} = [0.1520, 0.1560]$$

$$\mathcal{T}^{(2,3)} = [0.3600, 0.3960]$$

$$\mathcal{T}^{(3,4)} = [0.6520, 0.6560]$$

$$\mathcal{T}^{(4,1)} = [0.9020, 0.9120]$$

Figure 11: Performance validation tree for periodic scheduling example: $\vartheta_{out} = 0.8$, $\gamma = 0.09$

that if the controller is initialized with agent $K^{(1)}$ in the feedback loop, by the end of a single reference trajectory period, agent $K^{(1)}$ will have been switched back into the loop. Therefore, in this example, there is only a single switching cycle to be analyzed.

The periodic behavior of the switched system is seen in the timed transition table which can be derived from the performance validation tree. This is shown in figure 12. For the timed automaton $\mathcal{V}$ consists of four states with connecting directed arcs, $\mathcal{A}$, as indicated in the figure. The initial marking vector satisfies $\mu(1) = 1$. There are two local clocks, t_{init} and t. The clock t_{init} is never reset and controls only a single switch. The clock t is reset on every firing. Arcs are labeled with the reset constraints and the firing constraints, which are derived from the sets $\mathcal{T}^{(l,m)}$ with the reset condition taken into account (as described earlier).

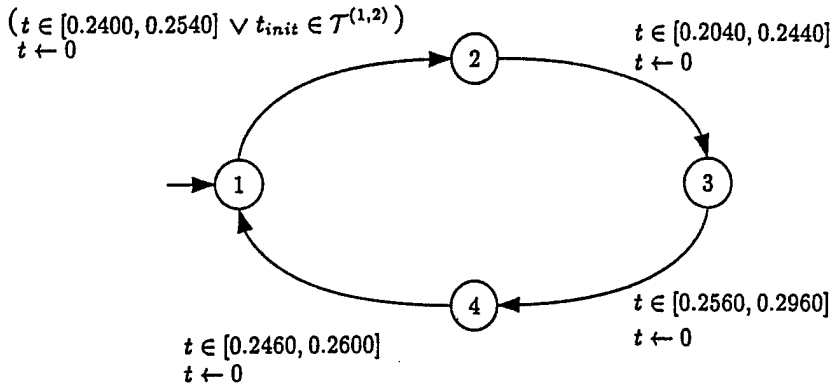


Figure 12: Timed transition table for periodic scheduling example: $\vartheta_{out} = 0.8$, $\gamma = 0.09$.

7 Conclusions

This paper has described an approach for analyzing the performance of switched LPV systems required to meet a bounded amplitude performance constraint with respect to a known scheduling trajectory. The LPV systems considered in this paper cover a large class of nonlinear systems which are driven or scheduled along a predetermined path of operating points which may or may not be states of the system.

The central theoretical result of the paper is the LPV Switching Lemma which states sufficient conditions for a system switched between two LPV realizations to satisfy the amplitude performance constraint over a given time interval. Using this result along with knowledge of the scheduling trajectory to compute bounds on the switching times, switching sequences can be checked against the LPV switching lemma to establish performance over a sequence of intervals which can be pieced together to establish performance over the entire reference scheduling trajectory. It was shown by example that the performance constraints can be verified computationally for finite-time and periodic scheduling trajectories.

In addition, this paper has described a method for extracting logical models representing the behavior of a class of scheduled continuous-time systems controlled by switching between a finite set of continuous-time controllers. The results of this paper show that knowledge of the system scheduling can be combined with robustness properties of LPV systems to derive logical models of the system behavior in the form of timed-automatons. The results in this paper focus on bounded amplitude performance condition, but there does not appear to be any reason prohibiting the use of these ideas for other performance problems, e.g. $\mathcal{H}_2$, $\mathcal{H}_\infty$. These results are therefore useful for the study of hybrid systems because they provide link between two distinctly different approaches to hybrid system design and analysis.

References

[1] R. Alur, C. Courcoubetis, T.A. Henzinger, and P.-H. Ho. Hybrid automata: an algorithmic approach to the specification and verification of hybrid systems. In R.L. Grossman, A. Nerode, A.P. Ravn, and H. Rischel, editors, *Hybrid Systems*, pages 209–229. Springer-Verlag, 1993. Lecture Notes in Computer Science 736.

[2] R. Alur and D.L. Dill. A theory of timed automata. *Theoretical Computer Science*, 126:183–235, 1994.

[3] P. Apkarian and P. Gahinet. A convex characterization of gain-scheduled H_∞ controllers. *IEEE Transactions on Automatic Control*, 40(5):853–864, May 1995.

[4] G. Becker and A. Packard. Robust performance of linear parametrically varying systems using parametrically-dependent linear feedback. *Systems and Control Letters*, 23:205–215, 1994.

[5] Christopher J. Bett and Michael Lemmon. Finite-horizon induced-L_∞ performance of linear parameter varying systems. In *Proceedings of the American Control Conference*, Albuquerque, NM, June 1997.

[6] C.J. Bett and M.D. Lemmon. Bounded amplitude control using multiple linear agents. Technical Report ISIS-97-004, Department of Electrical Engineering, University of Notre Dame, Notre Dame, IN, 1997.

[7] C.J. Bett and M.D. Lemmon. Finite horizon bounded amplitude control of linear parameter varying systems. Technical Report ISIS-97-002, Department of Electrical Engineering, University of Notre Dame, Notre Dame, IN, 1997.

[8] M. S. Branicky. Multiple Lyapunov functions and other analysis tools for switched and hybrid systems. *IEEE Transactions on Automatic Control*, to appear.

[9] J.R. Burch, E.M. Clarke, K.L. McMillan, D.L. Dill, and Hwang. Symbolic model checking: 10^{20} states and beyond. In *Proceedings of the Fifth Annual Symposium on Logic in Computer Science*. June 1990.

[10] M. Chilali and P. Gahinet. H_∞ design with pole placement constraints: an LMI approach. *IEEE Transactions on Automatic Control*, 41(3):358–367, March 1996.

[11] E.M. Clarke and E.A. Emerson. Characterizing properties of algorithms as fixed points. In *Seventh International Colloquim on Automata, Languages and Programming*. Springer-Verlag, 1981. Lecture Notes in Computer Science 85.

[12] M. Johansson and A. Rantzer. Computation of piecewise quadratic lyapunov functions for hybrid systems. Technical report, Lund Institute of Technology, Dept. of Automatic Control, 1996. to appear in *Transactions on Automatic Control*, special issue on Hybrid Systems.

[13] Edward W. Kamen and Pramod P. Khargonekar. On the control of linear systems whose coefficients are functions of parameters. *IEEE Transactions on Automatic Control*, AC-29(1):25–33, January 1984.

[14] H. K. Khalil. *Nonlinear Systems*. Prentice-Hall, Inc, 2nd edition, 1996.

[15] M. Lemmon and C. Bett. Extracting stable timed automata from switched model reference systems. In *Record of the Fifth International Hybrid Systems Workshop*, pages 270–282, Notre Dame, IN, September 1997.

[16] K. McMillan. Using unfoldings to avoid the state explosion problem in the verification of asynchronous circuits. In B.v. Bochmann and D.K. Probst, editors, *Computer Aided Verification, fourth international workshop, CAV'92*, pages 164–177. Springer-Verlag, 1992. Lecture Notes in Computer Science 663.

[17] A. S. Morse. Supervisory control of families of linear set-point controllers - part 1: Exact matching. *IEEE Transactions on Automatic Control*, 41(10):1413–1431, October 1996.

[18] Kameshwar Poolla and Jeff. S. Shamma. Optimal asymptotic robust performance via nonlinear controllers. *International Journal of Control*, 62(6):1367–1389, December 1995.

[19] Jeff S. Shamma and Michael Athans. Analysis of gain scheduled control for nonlinear plants. *IEEE Transactions on Automatic Control*, 35(8):898–907, 1990.

[20] Jeff S. Shamma and Michael Athans. Guaranteed properties of gain scheduled control for linear parameter-varying plants. *Automatica*, 27(3):559–564, 1991.

[21] Jeff S. Shamma and Michael Athans. Gain scheduling: Potential hazards and possible remedies. *Control Systems Magazine*, 12(3):101–107, 1992.

[22] J. Stiver, P.J. Antsaklis, and M. Lemmon. A logical DES approach to the design of hybrid systems. *Mathematical and Computer Modelling*, 10(8):??, 1996.

A Proofs

Proof of Proposition 4.1: By the assumption of the proposition, one can use arguments analogous to those used in the proof of theorem 1 in [7] to define functions $V_1 : \mathbb{R}^n \to \mathbb{R}$ and $V_2 : \mathbb{R}^n \to \mathbb{R}$ by

$$V_1(\xi) := \xi' P_1 \xi \quad V_2(\xi) := \xi' P_2 \xi \tag{65}$$

so that for any $\theta \in \mathcal{F}_C$ and along any trajectories of the switched system, the time derivatives of V_1 and V_2 must satisfy

$$\frac{d}{dt} V_1(x(t)) \leq -2\beta V_1(x(t)) \leq 0 \tag{66}$$

for any $t \in [0, t_s]$ and any $x(t)$ and $w(t)$ satisfying $x'(t)P_1 x(t) \geq r$ and $w'(t)w(t) \leq 1$ and

$$\frac{d}{dt} V_2(x(t)) \leq 0 \tag{67}$$

for any $t \in (t_s, T]$ and any $x(t)$ and $w(t)$ satisfying $x'(t)P_2 x(t) \geq 1$ and $w'(t)w(t) \leq 1$.

Given that $\beta > 0$, for any $\theta \in \mathcal{F}_C$ and $w \in \mathcal{BL}_\infty^{n_w}$, equation 66 implies that along system trajectories for $t \in [0, t_s]$, that

$$V_1(x(t)) \leq V_1(x(0)) + \int_0^t -2\beta V_1(x(\tau))d\tau \tag{68}$$

so that by the Bellman-Gronwall lemma

$$V_1(x(t)) \leq V_1(x(0)) \exp(-2\beta t) \tag{69}$$

Supposing that $V_1(x(0)) \leq 1$, the last equation implies $V_1(x(t)) \leq \exp(-2\beta t) \leq 1$ for all $t \in [0, t_s)$. If $t_s > t_d$, then $V_1(x(t)) \leq r$ for all $t > t_d$. Since $z(t) = C_1 x(t)$ it can be shown that equations 24 and 69 imply $z'(t)z(t) \leq \gamma^2$ for all $t \in [0, t_s]$.

The state trajectory is continuous at the switch so that $x(t_s^+) = x(t_s)$ which implies that $V_1(t_s^+) \leq r$. Combining this fact with equation 23 implies that $V_2(t_s^+) \leq 1$. Since $z(t) = C_2 x(t)$ for $t > t_s$ and since C_2 is a constant matrix, one has $z(t_s^+) = C_2 x(t_s^+)$ so that equations 25 and 27 imply $z'(t_s^+)z(t_s^+) \leq \gamma^2$.

Now suppose that $V_2(x(t)) > 1$ for $t \in (t_s, T]$. This implies that either $V_2(x(t_s^+)) \geq 1$ and $\dot{V}_2(x(t_x^+)) > 0$ with $w'(t_s^+)w(t_s^+) \leq 1$ or it implies that there exists a $\tau \in (t_s, T)$ such that $V_2(x(\tau)) \geq 1$ and $\dot{V}_2(x(\tau)) > 0$ with $w'(\tau)w(\tau) \leq 1$. Since $\theta(t_s^+) \in \Theta_2$ and since equation 67 holds for any $\theta \in \mathcal{F}_C$ then $\dot{V}(x(t_s^+)) < 0$ which generates a contradiction. The only conclusion is that $V_2(x(t)) \leq 1$ for all $t \in (t_s, T]$. Since $z(t) = C_2 x(t)$ in this time interval, one immediately concludes that $z'(t)z(t) \leq \gamma^2$ for all $t \in (t_s, T]$. The result follows immediately. □

Proof of Lemma 5.1: Under the assumptions of the lemma and from the proof of theorem 1 in [7], for any $\theta \in \mathcal{F}_\Theta[0,T]$

$$\|z(t)\| \leq \gamma \max\{\sqrt{r}, e^{-\beta t}\} \tag{70}$$

for all t. By assumption,

$$|\theta_i(t) - \theta_{nom,i}(t)| = |S_i(x_m(t), x(t), v(t)) - S_i(x_m(t), 0, 0)| \tag{71}$$

for $i = 1, 2, \ldots, s$. The assumptions on the parameter mapping therefore imply that

$$|S_i(x_m(t), x(t), v(t)) - S_i(x_m(t), 0, 0)| \leq k_{zi}\gamma \max\{\sqrt{r}, e^{-\beta t}\} \tag{72}$$

Combining these last two equations yields the result. □

Proof of Corollary 5.1: To prove the right-hand inequality, note that from lemma 5.1

$$|\theta_i(T) - \theta_{nom,i}(T)| \leq k_{zi}\gamma \max\{\sqrt{r}, e^{-\beta T}\} \tag{73}$$

for $i = 1, 2, \ldots, s$. Adding this to equation 33 yields

$$|\bar{\theta}_i - \theta_i(T)| + |\theta_i(T) - \theta_{nom,i}(T)| \leq \vartheta_{out} + k_{zi}\gamma \max\{\sqrt{r}, e^{-\beta T}\}$$

for $i = 1, 2, \ldots, s$. Applying the triangle inequality yields

$$|\bar{\theta}_i - \theta_{nom,i}(T)| \leq \vartheta_{out} + k_{zi}\gamma \max\{\sqrt{r}, e^{-\beta T}\}$$

To prove the left-hand inequality, write

$$|\bar{\theta}_i - \theta_{nom,i}(T)| \geq |\bar{\theta}_i - \theta_i(T)| - |\theta_i(T) - \theta_{nom,i}(T)|$$

Combining with equations 33 and 73 yields the desired result. □

Proof of Proposition 5.1: Let $\tau = t_s - t_0$. Under the assumptions of the proposition, if $\hat{x}'(t_0)P\hat{x}(t_0) \leq 1$ and $w \in \mathcal{BL}_\infty$, lemma 5.1 implies that

$$|\theta_i(t_s) - \theta_{nom,i}(t_s)| \leq k_{zi}\gamma \max\{\sqrt{r}, e^{-\beta \tau}\} \tag{74}$$

If the parameter trajectory satisfies the nearest neighbor switching rule, then at t_s,

$$\|\theta(t_s) - \theta_{nom}^{(m)}\| \leq \|\theta(t_s) - \theta_{nom}^{(q)}\| \tag{75}$$

for all $q \in I_K$ with q not equal to m. This implies that $\theta(t_s) \in \hat{\Theta}_m$ so that t_s must satisfy the first condition for $\mathcal{T}^{(l,m)}$. For the second condition, from corollary 5.1, $\theta_{nom}(t_s)$ must satisfy

$$\vartheta_{out} - k_{zi}\gamma \max\{\sqrt{r}, e^{-\beta \tau}\} \leq \left|\theta_{nom,i}(t_s) - \theta_{nom,i}^{(l)}\right| \leq \vartheta_{out} + k_{zi}\gamma \max\{\sqrt{r}, e^{-\beta \tau}\} \tag{76}$$

for some $i = 1, 2, \ldots, s$. One concludes that $t_s \in \mathcal{T}^{(l,m)}$. □

Proof of Proposition 5.2: Consider an admissible switching sequence $k \to l \to m$ with $\alpha_1^{(i)}$, $\alpha_2^{(i)}$ and $P^{(i)}$ satisfying (39-40). Define t_i as the time that control agent $K^{(i)}$ is switched into feedback with the system, $i \in \{k, l, m\}$. Assume, without loss of generality, that $\hat{x}(t_k)' P^{(k)} \hat{x}(t_k) \leq 1$. To apply the LPV switching lemma to prove performance over the interval $[t_k, t_m]$, it must be established that $\theta(t)$ is legal and that the dwell-time constraint is satisfied.

To demonstrate the legality of $\theta(t)$, note that from the integrator controller structure and continuity of the parameter mapping, $\theta(t)$ must be continuous. By the nearest neighbor switching rule and adequate sampling assumption, $\theta(t) \in \mathcal{F}_C$ (i.e. $\theta(t)$ is legal).

To demonstrate the dwell-time constraint, note that from proposition 5.1, $\min \mathcal{T}^{(l,m)} - \max \mathcal{T}^{(k,l)}$ is a lower bound on $t_l - t_k$. The dwell-time requirement is satisfied if there exists an $r \in \mathcal{R}^{(l,m)}$ such that

$$f(r) \leq \min \mathcal{T}^{(l,m)} - \max \mathcal{T}^{(k,l)}$$

where

$$f(r) = \frac{r}{\alpha_2^{(l)} - \alpha_1^{(l)} r} \log r$$

and $\mathcal{R}^{(l,m)}$ is as defined in (45). It is easily shown that $f(r)$ is monotonically decreasing on $\mathcal{R}^{(l,m)}$. Thus, (41) implies that the dwell-time requirement is satisfied for $t_l - t_k$. By the LPV switching lemma,

$$\|z(t)\| \leq \gamma, \quad t \in [t_k, \tau_l)$$

for any $\tau_l \in [t_l, t_m)$. Furthermore, $\hat{x}(t_l)' P^{(l)} \hat{x}(t_l) \leq 1$.

As a consequence of the nearest neighbor switching rule, the adequate sampling assumption and the continuity of $\theta(t)$, any finite time interval, $[0, T]$, can contain at most a finite number of switching instants. Assume, without loss of generality, $N + 1$ agents switched in order $0, 1, 2, \ldots, N$. Thus, the interval $[0, T]$ can be written

$$[0, T] = \bigcup_{i=0}^{N-1} [t_i, \tau_{i+1}) + [t_N, T]$$

where $t_0 = 0$ and the interval $[t_i, \tau_{i+1})$ denotes the the admissible switching sequence $i \to i+1 \to i+2$ with $\tau_i \in [t_i, t_{i+1}]$ for $i = 1, \ldots, N-1$ and $\tau_N \in [t_N, T]$. By assumption, $\hat{x}(t_0)' P^{(0)} \hat{x}(t_0) \leq 1$ so that with the LPV switching lemma, one concludes that

$$\|z(t)\| \leq \gamma, \quad t \in [t_0, \tau_1) \quad \text{and} \quad \hat{x}(t_1)' P^{(1)} \hat{x}(t_1) \leq 1.$$

In similar fashion,

$$\|z(t)\| \leq \gamma, \quad t \in [t_1, \tau_2) \quad \text{and} \quad \hat{x}(t_2)' P^{(2)} \hat{x}(t_2) \leq 1.$$

By induction, one concludes that $\|z(t)\| \leq \gamma$ for subintervals $t \in [t_i, \tau_{i+1})$, $i = 1, \ldots, N-1$, and $\hat{x}(t_N)' P^{(N)} \hat{x}(t_N) \leq 1$. Since (39-40) is satisfied for $i = N$ and since $\theta(t) \in \mathcal{F}_\Theta[t_N, T]$ by the fact that there is no other switch according to the nearest neighbor switching rule, $\|z(t)\| \leq \gamma$ for $t \in [t_N, T]$ as well. One concludes that performance is satisfied over the finite interval,

$$\|z\|_{\infty,[0,T]} \leq \gamma.$$

For $T \to \infty$, there are two possibilities: switching stops after a finite time, or switching does not stop. In the first case, there are a finite number of switches and performance is proven by the above analysis. In the second case, an infinite number of switches must be considered. In that case,

$$[0, \infty) = \bigcup_{i=0}^{\infty} [t_i, \tau_{i+1})$$

Define

$$\bar{z}_i := \operatorname{ess} \sup_{t \in [t_i, \tau_{i+1})} \|z(t)\|$$

and $\mathcal{Z} := \{z \mid \|z\| \leq \gamma\}$ as all points z satisfying performance constraints. Clearly, from the preceding analysis, $\bar{z}_i \in \mathcal{Z}$. Furthermore, any limit point of the sequence $\{\bar{z}_i\}$ must lie in the closure of $\mathcal{Z}$. Since $\mathcal{Z}$ is closed and bounded, $\bar{z} \in \mathcal{Z}$, implying that performance is satisfied in the case of an infinite number of switches. Thus,

$$\|z\|_{\infty,[0,\infty)} \leq \gamma.$$

□

B Choosing r and β in Lemma 5.1

Lemma 5.1 is important because it implies that if the bound on the parameter trajectory is small enough, then the nominal parameter trajectory represents a reasonable approximation to the true parameter trajectory. Unfortunately, the parameters r and β required to evaluate the bounds are not specified precisely and it is not immediately clear how these values should be chosen from the feasible solutions of inequalities 30 and 31. Note that r and β appear in the term $\max\{\sqrt{r}, \exp(-\beta t)\}$ in equation 32; to obtain the best bound at a given time t, r and β should be chosen to minimize $\max\{\sqrt{r}, \exp(-\beta t)\}$.

Now consider the requirements of lemma 5.1 rewritten to require constants $\alpha_1 \geq \alpha_2 > 0$ and a positive definite matrix P such that

$$P \geq \frac{1}{\gamma^2} C'C \tag{77}$$

and

$$P \in \text{FeasRic}\left(A(\theta) + B_u K, B_w(\theta), \alpha_1, \alpha_2\right) \tag{78}$$

for all $\theta \in \Theta$ where

$$\alpha_2 = \alpha \qquad \text{and} \qquad \alpha_1 = 2\beta + \frac{\alpha_2}{r}.$$

It is clear that for any α_1 and α_2 which satisfy inequalities 77 and 78 for some positive definite matrix P, any r and β satisfying

$$\beta = \frac{1}{2}\left(\alpha_1 - \frac{\alpha_2}{r}\right), \qquad r \in \left[\frac{\alpha_2}{\alpha_1}, 1\right] \tag{79}$$

are also feasible solutions of inequalities 30 and 31 for the same positive definite matrix P. For fixed α_1 and α_2 which satisfy the above constraints, substituting the expression for β in equation 79 into $\max\{\sqrt{r}, \exp(-\beta t)\}$ yields the optimization problem

$$\min_{r \in \left[\frac{\alpha_2}{\alpha_1}, 1\right]} \max \left\{\sqrt{r}, \exp(-\frac{1}{2}(\alpha_1 - \frac{\alpha_2}{r})t)\right\} \tag{80}$$

which is depicted graphically in figure 13. Figure 13 clearly shows that the optimization problem is solved by finding the value of r which satisfies

$$r = \exp\left(-\left(\alpha_1 - \frac{\alpha_2}{r}\right)t\right) \tag{81}$$

Although there is no analytical solution for r in terms of α_1, α_2 and t, numerical solutions are easily obtained. (For details of one possible approach, see [6].) Note that figure 13 provides guidelines for choosing α_1 and α_2. Given a collection of feasible points which satisfy the inequalities 77 and 78, the pair α_1 and α_2 should be chosen to maximize $\alpha_1 - \alpha_2$. Intuitively, this should reduce the optimal cost of the problem in equation 80 by forcing the intersection point in figure 13 to occur at a lower value of r.

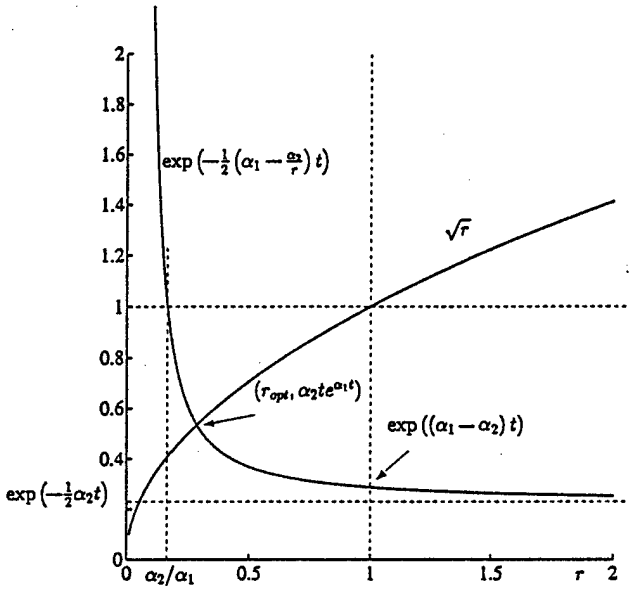


Figure 13: Choosing r and β

APPENDIX E

C.J. Bett and M.D. Lemmon
"Sufficient Conditions for Self-Scheduled Bounded Amplitude Control"
Submitted to *Automatica* September 1997
Also released as ISIS Technical Report ISIS-97-009, University of Notre Dame

Sufficient Conditions for Self-Scheduled Bounded Amplitude Control

Technical Report of the ISIS Group

at the University of Notre Dame

ISIS-97-009

September 1997

Christopher J. Bett and Michael Lemmon

Department of Electrical Engineering

University of Notre Dame

Notre Dame, IN 46556

Interdisciplinary Studies of Intelligent Systems

Sufficient Conditions for Self-Scheduled Bounded Amplitude Control

Christopher J. Bett and Michael D. Lemmon*
Department of Electrical Engineering
University of Notre Dame
Notre Dame, IN 46556

September 1997

Abstract

This paper considers the problem of synthesizing a parameter-dependent output feedback controller for continuous-time linear parameter varying (LPV) systems so that a prespecified bound on signal amplitude is satisfied by the controlled system. The main result of the paper is a sufficient condition for the existence of an LPV controller which guarantees an upper bound on the induced-$\mathcal{L}_\infty$ norm of the controlled system. The condition takes the form of linear matrix inequalities (LMIs) which must be solved pointwise over a parameter set. A set of LPV controllers which satisfy the performance constraints are parameterized in terms of feasible solutions to the LMI existence conditions. In the event that the parameter variations are removed, the conditions reduce to necessary and sufficient LTI synthesis conditions for bounded amplitude performance. The existence conditions are amenable to efficient numerical techniques.

1 Introduction

A measure of system performance which frequently arises in control problems is the *peak value* of an appropriately selected plant signal. Control problems in which this performance measure arise are prevalent in the literature: motor control problems with electrical (voltage or current) or mechanical (motion) restrictions and process control problems with chemical concentration restrictions are two examples. In each of these examples, violation of the amplitude restrictions can lead to performance degradation and possibly catastrophic system failure. Furthermore, energy-based design techniques, such as the $\mathcal{H}_\infty$ design paradigm, are often inadequate for these types of performance problems.

In this paper, we consider the problem of synthesizing controllers for finite-dimensional linear parameter varying (LPV) plants so that the controlled system satisfied bounded amplitude performance constraints. LPV systems are linear systems whose state-space matrices depend in a continuous fashion on a time-varying parameter vector, $\theta(t) \in \mathbb{R}^s$. While the trajectory of $\theta(t)$ is not known *a priori*, it is assumed to

*The authors gratefully acknowledge the partial financial support of the Army Research Office (DAAH04-95-1-0600, DAAH04-96-1-0134). e-mail:lemmon@maddog.ee.nd.edu

be measurable in real-time, thus providing real-time information on the behavior of the system. Intuition alone suggests that incorporating such information into the controller should improve overall performance of the controlled system since such information allows the controller to adjust, or *schedule* to the change in operating conditions $\theta(t)$. A controller which incorporates real-time parameter measurements in this fashion is called a *self-scheduled controller.* Such controllers are justified empirically by the widely successful gain-scheduling technique.

The relationship between classical gain-scheduling and parameter-dependent systems is seen in [13] and [14]. Early work on parameter-dependent controllers is seen in [10]; more recently, parameter-dependent synthesis approaches based on Lyapunov and small-gain techniques have appeared [12][5][2] which frame the synthesis problem as a convex optimization problem. These approaches generalize $\mathcal{H}_\infty$ techniques for LTI systems to LPV systems. We note that while this list is not exhaustive, current literature on self-scheduled control is limited in scope to stability and $\mathcal{L}_2$, or $\mathcal{H}_\infty$, performance problems.

In fact, bounded amplitude control is a difficult problem even for systems with no parameter dependence (i.e. LTI systems). While bounded amplitude control of discrete-time linear systems is well-understood [15][8], bounded amplitude control of continuous-time systems remains a difficult task [9]. A relatively new approach to this problem for deterministic, linear time-invariant systems was recently proposed in [11]. At the heart of this approach is the identification of an invariant ellipsoid which bounds the reachable states of the system under the assumption of an amplitude-bounded exogenous input. The ellipsoid is characterized by a positive definite solution to a Riccati matrix inequality coupled with an additional linear matrix inequality (LMI) [7]. This approach was extended in [6] to the synthesis of constant gain state-feedback controllers for LPV systems.

In this paper, we generalize the results of [6] and [1] to the synthesis of parameter-dependent controllers for the bounded amplitude performance problem. The approach presented in this paper is similar in nature to the parameter-dependent controller results presented in [5]. The primary difference between this new result is the nature of the performance problem. In [5], results are presented for an $\mathcal{L}_2$ ($\mathcal{H}_\infty$) performance problem; here, we consider a bounded amplitude, or induced-$\mathcal{L}_\infty$ performance problem. As with other self-scheduling, LTI synthesis results are recovered when parameter dependence is removed (i.e. the state-space matrices are constant). The derivation presented here follows that presented in [11] for bounded amplitude LTI performance with notable exceptions. First, and most apparent, the problem considered here deals with a more general class of systems, namely, LPV systems. Because of this, the algebraic Riccati equation used

in the LTI synthesis are no longer useful. The ARE must be relaxed to an appropriate Riccati inequality, much as was done in the $\mathcal{L}_2$ self-scheduling problem.

The remainder of this paper is organized as follows. Section 2 presents the mathematical notation used throughout the paper. Section 3 presents some preliminary results which characterize bounded amplitude performance of LPV systems and states the output feedback synthesis results for LTI systems. The main results concerning synthesis of parameter dependent control for LPV systems are located in section 4. Computation issues are discussed in section 5.

2 Mathematical Background

Many of the definitions for LPV systems presented in this section are generalized from [5] and the references therein.

Definition 2.1 *The infinity norm of a signal* $f : \mathbb{R}^+ \to \mathbb{R}^n$ *is defined as*

$$\|f\|_\infty := \operatorname{ess\,sup}_{t \geq 0} \|f(t)\|$$

where $\|\cdot\|$ *denotes the Euclidean vector norm.* $\mathcal{L}_\infty^n$ *denotes the normed* n*-dimensional signal space in the usual fashion under the above signal norm definition;* $\mathcal{BL}_\infty^n$ *denotes the subset of* $\mathcal{L}_\infty^n$ *with signal norms bounded above by unity.*

Definition 2.2 (Parameter Variation Set) *Given a compact subset* $\Theta \subset \mathbb{R}^s$*, the* parameter variation set $\mathcal{F}_\Theta$ *denotes the set of all continuous functions mapping* $\mathbb{R}^+$ *into* Θ*.*

The notation $\theta \in \mathcal{F}_\Theta$ denotes a function in the parameter variation set; $\theta \in \Theta$ denotes a vector in a compact subset of $\mathbb{R}^s$. Note that both $\mathcal{F}_\Theta$ and, for instance, $\mathcal{L}_\infty^n$ represent signal spaces. Technically, $\mathcal{F}_\Theta \subset \mathcal{L}_\infty^s$ since $\mathcal{F}_\Theta$ consists of supremum bounded s-dimensional vectors which vary continuously in time. In this paper, $\mathcal{F}_\Theta$ will always refer to parameter signals or parameter variations; $\mathcal{L}_\infty^n$ will refer to signals in the plant input, output or state space.

Definition 2.3 (LPV System) *Given a compact set* $\Theta \subset \mathbb{R}^s$*, and continuous functions* $A : \mathbb{R}^s \to \mathbb{R}^{n \times n}$*,* $B : \mathbb{R}^s \to \mathbb{R}^{n \times n_w}$*,* $C : \mathbb{R}^s \to \mathbb{R}^{n_z \times n}$*, and* $D : \mathbb{R}^s \to \mathbb{R}^{n_z \times n_w}$*, an* n^{th} *order* linear parameter varying (LPV) system *is a dynamical system whose dynamics evolve as*

$$\begin{bmatrix} \dot{x}(t) \\ z(t) \end{bmatrix} = \begin{bmatrix} A(\theta(t)) & B(\theta(t)) \\ C(\theta(t)) & D(\theta(t)) \end{bmatrix} \begin{bmatrix} x(t) \\ w(t) \end{bmatrix} \tag{1}$$

where $\theta \in \mathcal{F}_\Theta$.

The performance measure used in this paper is the induced-$\mathcal{L}_\infty$ norm which, as defined below, is related to the peak-to-peak gain of the system.

Definition 2.4 *Let the LPV system of definition 2.3 be denoted by* $\Sigma(\Theta, A, B, C, D)$. *For any* $\theta \in \mathcal{F}_\Theta$, *the linear time-varying system described in equation 1 is denoted* Σ_θ. *The state-transition matrix of* Σ_θ *is denoted* $\Phi_\theta(t, t_0)$. *For* $x(t_0) = 0$, *the causal linear input/output mapping,* $H_\theta : \mathcal{L}_\infty^{n_w} \to \mathcal{L}_\infty^{n_z}$, *of* Σ_θ *is defined as*

$$H_\theta w(t) \quad = \quad \int_{t_0}^{t} C(\theta(\tau))\Phi_\theta(t,\tau)B(\theta(\tau))w(\tau)d\tau + D(\theta(t))w(t)$$

The induced-$\mathcal{L}_\infty$ *norm of* H_θ *is given by*

$$\|H_\theta\|_{i\infty} := \sup_{w \in \mathcal{B}\mathcal{L}_\infty^{n_w}} \|H_\theta w\|_\infty. \tag{2}$$

The shorthand $\|H_{\mathcal{F}_\Theta}\|_{i\infty} \leq \gamma$ *for some* $\gamma > 0$ *means that for all* $\theta \in \mathcal{F}_\Theta$, $\|H_\theta\|_{i\infty} \leq \gamma$ *where* $H_{\mathcal{F}_\Theta} := \{H_\theta : \theta \in \mathcal{F}_\Theta\}$.

Finally, consider an LPV plant input/output mapping, P_θ, which maps input vectors $[w'\ u']'$ to output vectors $[z'\ y']'$ and LPV feedback control mapping, K_θ, which maps y to u. The closed loop mapping from w to z will be denoted $\mathcal{F}_\ell(P_\theta, K_\theta)$.

Remark: When the parameter set Θ is a singleton, parameter variations are fixed at a single point. In this case, all of the above definitions reduce to the corresponding notions for LTI systems.

3 Preliminary Results

This section contains preliminary results establishing conditions for bounded amplitude performance of LPV systems and stating some previously derived results[1] for synthesis of output feedback control which maintains a bound on the induced-$\mathcal{L}_\infty$ norm of LTI systems. These results all concern strictly proper systems. Some of the notation for this section is borrowed from [1].

3.1 Bounded Amplitude Performance

Consider an LPV system, H, with state-space realization

$$H(\theta) \quad := \quad \left[\begin{array}{c|c} A(\theta) & B(\theta) \\ \hline C(\theta) & 0 \end{array}\right]. \tag{3}$$

defined for a parameter set Θ. The following result is a minor variation of the result proven in [6]; the proof is omitted.

Theorem 3.1 *Fix $\alpha > 0$ and performance level $\gamma > 0$ and consider the LPV system of (3) with $(A(\theta), B(\theta))$ controllable for all $\theta \in \Theta$. If there exists a matrix $Q > 0$ such that*

$$A(\theta)Q + QA'(\theta) + \alpha Q + \frac{1}{\alpha}B(\theta)B'(\theta) \leq 0 \tag{4}$$

$$C'(\theta)C(\theta) < \gamma^2 Q^{-1} \tag{5}$$

for all $\theta \in \Theta$, then

$$\|H_{\mathcal{F}_\Theta}\|_{i\infty} < \gamma$$

For a fixed α, let the set V_α be the set of all γ such that the matrix inequalities (4)-(5) admit a solution, i.e.

$$V_\alpha := \left\{ \gamma \;\middle|\; \exists Q > 0 \text{ such that } A(\theta)Q + QA'(\theta) + \alpha Q + \frac{1}{\alpha}B(\theta)B'(\theta) \leq 0,\ C'(\theta)C(\theta) < \gamma^2 Q^{-1}\ \forall \theta \in \Theta \right\} \tag{6}$$

V_α is the set of all upper bounds on $\|H_{\mathcal{F}_\Theta}\|_{i\infty}$ corresponding to a fixed parameter α; the smallest of these will be denoted

$$N_\alpha(H_{\mathcal{F}_\Theta}) := \begin{cases} \inf V_\alpha, & \alpha \in (0, \kappa) \\ \infty & \alpha \geq \kappa \end{cases} \tag{7}$$

where

$$\kappa := -2 \max_{\theta \in \Theta} \max_{i=1,\ldots,n} (\Re(\lambda_i(A(\theta)))),$$

is the smallest upper bound corresponding to α when V_α is nonempty. Here, $\lambda_i(A(\theta))$ denotes the i^{th} eigenvalue of $A(\theta)$ for some $\theta \in \Theta$.

Remark: $N_\alpha(H_{\mathcal{F}_\Theta}) := \infty$ for $\alpha \geq \kappa$ for the following reason. (4) can be written as

$$(A(\theta) + \frac{1}{2}\alpha I)Q + Q(A(\theta) + \frac{1}{2}\alpha I)' + \frac{1}{\alpha}B(\theta)B'(\theta) \leq 0.$$

For $\alpha \geq \kappa$, there exists a $\theta \in \Theta$ such that $A(\theta) + \frac{1}{2}\alpha I$ is no longer Hurwitz, so no $Q > 0$ can satisfy (4) for this θ.

The following corollary to theorem 3.1 is now apparent.

Corollary 3.1 *Fix $\alpha > 0$ and performance level $\gamma > 0$ and consider the LPV system of (3) with $(A(\theta), B(\theta))$ controllable for all $\theta \in \Theta$. Then for all $\theta \in \mathcal{F}_\Theta$, $N_\alpha(H_{\mathcal{F}_\Theta}) < \gamma$ if and only if there exists a matrix $Q > 0$ such that (4)-(5) admit a solution for all $\theta \in \Theta$.*

Clearly, from the corollary,

$$\|H_{\mathcal{F}_\Theta}\|_{i\infty} \leq \inf_{\alpha>0} N_\alpha(H_{\mathcal{F}_\Theta})$$

Thus, corollary 3.1 generalizes the LTI results of [11].

3.2 LTI System Results

In the case when Θ is a singleton, the LPV system may be treated as an LTI system. Output feedback synthesis results for LTI systems were previously derived in [1]. Some of these results are restated here so that they may be applied in the next section.

Consider an LTI system, P, with state space realization

$$\begin{bmatrix} \dot{x}(t) \\ z(t) \\ y(y) \end{bmatrix} = \begin{bmatrix} A & B_1 & B_2 \\ C_1 & 0 & D_{12} \\ C_2 & D_{21} & 0 \end{bmatrix} \begin{bmatrix} x(t) \\ w(t) \\ u(y) \end{bmatrix} \tag{8}$$

The system in (8) satisfies the following assumptions.

(LTI1) (A, B_1, C_2) controllable and detectable. This ensures that an appropriately defined Riccati equation has a unique positive definite solution.

(LTI2) $B_1 D_{21}' = 0$. This means that process noise $B_1 w$ is entirely decoupled from measurement noise $D_{12} w$.

(LTI3) D_{21} has full row rank. This means that measurement noise can corrupt all measurements.

The primary result of this section is stated in the following theorem.

Theorem 3.2 ([1]) *Fix any number $\alpha > 0$ and performance level γ and consider the system defined in (8) under assumptions LTI1-LTI3. The following are equivalent:*

1. *There exists a strictly proper, finite-dimensional,LTI controller K which internally stabilizes the system and renders $N_\alpha(\mathcal{F}_\ell(P,K)) < \gamma$.*
2. *The LMI (in $Z = Z'$ and V)*

$$\begin{bmatrix} AZ + ZA' + B_2V + V'B_2' + \alpha Z & B_1 \\ B_1' & -\alpha I \end{bmatrix} \leq 0 \tag{9}$$

$$\begin{bmatrix} (Z - Y_\alpha) & (Z - Y_\alpha)C_1' + V'D_{12}' \\ C_1(Z - Y_\alpha) + D_{12}V & \gamma^2 I - C_1 Y_\alpha C_1' \end{bmatrix} > 0 \tag{10}$$

admits a solution with $Z > 0$, where Y_α is the stabilizing solution to the algebraic Riccati equation

$$(A + \frac{1}{2}\alpha I)Y_\alpha + Y_\alpha(A + \frac{1}{2}\alpha I)' - \alpha Y_\alpha C_2'(D_{21}D_{21}')^{-1}C_2 Y_\alpha + \frac{1}{\alpha}B_1 B_1' = 0 \tag{11}$$

Moreover, if either (hence, both) of these statements hold, then one controller that renders $N_\alpha(\mathcal{F}_\ell(P,K)) < \gamma$ is given by

$$K = \left[\begin{array}{c|c} A + B_2\bar{J} - \alpha Y_\alpha C_2'(D_{21}D_{21}')^{-1}C_2 & \alpha Y_\alpha C_2'(D_{21}D_{21}')^{-1} \\ \hline \bar{J} & 0 \end{array}\right] \tag{12}$$

where $\bar{J} := V(Z - Y_\alpha)^{-1}$ and Z, V and Y_α satisfy the conditions of part 2.

Theorem 3.2 may be used to synthesis strictly proper LTI controllers which enforce an amplitude constraint on the controlled plant. The approach is to perform a line search for values of $\alpha > 0$, evaluating the conditions of the theorem at each point until specified performance, γ, is achieved.

4 Strictly Proper LPV Output Feedback

The plant description is now generalized to the LPV system, $P(\theta)$, with state space realization

$$\begin{bmatrix} \dot{x}(t) \\ z(t) \\ y(t) \end{bmatrix} = \begin{bmatrix} A(\theta) & B_1(\theta) & B_2(\theta) \\ C_1(\theta) & 0 & D_{12}(\theta) \\ C_2(\theta) & D_{21}(\theta) & 0 \end{bmatrix} \begin{bmatrix} x(t) \\ w(t) \\ u(t) \end{bmatrix} \tag{13}$$

defined over a parameter set Θ, which satisfies the following assumptions.

(LPV1) $(A(\theta), B_1(\theta), C_2(\theta))$ controllable and detectable for all $\theta \in \Theta$.

(LPV2) $B_1(\theta)D_{21}'(\theta) = 0$ for all $\theta \in \Theta$.

(LPV3) $D_{21}(\theta)$ has full row rank for all $\theta \in \Theta$.

To generalize the previous results to self-scheduled controllers for LPV plants, the Riccati equation must be relaxed to a matrix inequality. We make the following definition.

Definition 4.1 *Suppose that $(A(\theta), R(\theta))$ is stabilizable for all $\theta \in \Theta$. Let the mapping $\mathcal{Q} : \mathbb{R}^{n\times n} \to \mathbb{R}^{n\times n}$ be defined by*

$$\mathcal{Q}(Y) := A(\theta)Y + YA'(\theta) + YR(\theta)Y + M(\theta). \tag{14}$$

Suppose that $R(\theta) \leq 0$ *for all* $\theta \in \Theta$. *Then a positive definite matrix* Y_{min} *satisfying*

$$\mathcal{Q}(Y_{min}) \leq 0, \ \forall \theta \in \Theta$$

is called minimal stabilizing solution *if*

$$Y_{min} \leq Y \quad \textit{for all} \quad Y > 0, \ \mathcal{Q}(Y) \leq 0, \ \forall \theta \in \Theta.$$

Remark: When Θ is a singleton, e.g. $\Theta = \theta_0$, then the minimal stabilizing solution, in the sense defined above, is the stabilizing solution to the algebraic Riccati equation

$$A(\theta_0)Y_{min} + Y_{min}A'(\theta_0) + Y_{min}R(\theta_0)Y_{min} + M(\theta_0) \ = \ 0 \tag{15}$$

To see this, take any positive definite Y such that $\mathcal{Q}(Y) \leq 0$ for all $\theta \in \Theta$. Then

$$\mathcal{Q}(Y) - \mathcal{Q}(Y_{min}) \leq 0$$

Equivalently,

$$(A(\theta_0) + Y_{min}R(\theta_0))(Y - Y_{\min}) + (Y - Y_{\min})(A(\theta_0) + Y_{min}R(\theta_0))'$$
$$\alpha(Y - Y_{\min}) + (Y - Y_{\min})R(\theta_0)(Y - Y_{\min}) \leq 0$$

for all $\theta \in \Theta$. This implies the existence of a matrix $\hat{M}(\theta_0)$ so that

$$(A(\theta_0) + Y_{min}R(\theta_0))(Y - Y_{\min}) + (Y - Y_{\min})(A(\theta_0) + Y_{min}R(\theta_0))'$$
$$\alpha(Y - Y_{\min}) + (Y - Y_{\min})R(\theta_0)(Y - Y_{\min}) + \hat{M}(\theta_0) = 0 \tag{16}$$

Since Y_{min} is a stabilizing solution to the ARE in (15), $A(\theta_0) + Y_{min}R(\theta_0)$ is stable. This implies that $Y - Y_{\min} \geq 0$ (see, e.g. [17]), or $Y \geq Y_{min}$.

We are now ready to state the main result of the paper.

Theorem 4.1 *Fix any number* $\alpha > 0$ *and performance level* γ *and consider the LPV plant in (13) under assumptions LPV1-LPV3. Then a sufficient condition for the existence of a strictly proper, finite-dimensional, LPV controller* $K(\theta)$ *which internally stabilizes the system and for all* $\theta \in \mathcal{F}_\Theta$, *renders* $N_\alpha(\mathcal{F}_\ell(P(\theta), K(\theta))) < \gamma$ *is the existence of matrices* $Z = Z' > 0$ *and* V *such that*

$$\begin{bmatrix} A(\theta)(Z - Y_\alpha) + (Z - Y_\alpha)A'(\theta) + B_2(\theta)V + V'B_2'(\theta) + \alpha(Z - Y_\alpha) & Y_\alpha C_2'(\theta) \\ C_2(\theta)Y_\alpha & -\frac{1}{\alpha}D_{21}(\theta)D_{21}'(\theta) \end{bmatrix} \leq 0 \tag{17}$$

$$\begin{bmatrix} (Z - Y_\alpha) & (Z - Y_\alpha)C_1'(\theta) + V'D_{12}'(\theta) \\ C_1(\theta)(Z - Y_\alpha) + D_{12}(\theta)V & \gamma^2 I - C_1(\theta)Y_\alpha C_1'(\theta) \end{bmatrix} > 0 \quad (18)$$

for all $\theta \in \Theta$ where Y_α is the minimal stabilizing solution (in the sense of definition 4.1) of the Riccati inequality

$$(A(\theta) + \frac{1}{2}\alpha I)Y_\alpha + Y_\alpha(A(\theta) + \frac{1}{2}\alpha I)' - \alpha Y_\alpha C_2'(\theta)(D_{21}(\theta)D_{21}'(\theta))^{-1}C_2(\theta)Y_\alpha + \frac{1}{\alpha}B_1(\theta)B_1'(\theta) \leq 0 \quad (19)$$

In addition, if the above matrix inequalities admit a solution, then one controller that renders $N_\alpha(\mathcal{F}_\ell(P(\theta), K(\theta))) < \gamma$ for all $\theta \in \mathcal{F}_\Theta$ is given by

$$K = \left[\begin{array}{c|c} A(\theta) + B_2(\theta)\bar{J} - \alpha Y_\alpha C_2'(\theta)(D_{21}(\theta)D_{21}(\theta)')^{-1}C_2(\theta) & \alpha Y_\alpha C_2'(\theta)(D_{21}(\theta)D_{21}(\theta)')^{-1} \\ \hline \bar{J} & 0 \end{array}\right] \quad (20)$$

where $\bar{J} := V(Z - Y_\alpha)^{-1}$ and Z, V and Y_α satisfy the conditions of part 2.

Furthermore, if Θ is a singleton, then the above conditions are strengthened to necessary and sufficient.

Proof of Theorem 4.1: Suppose that the LMIs of equation 17 and 18 in variables Z and V admits a solution with $Z > 0$, and suppose the loop is closed with the realization given in equation 20. Let x and $\hat{x}$ denote the state of the plant and controller, respectively. Define $W(\theta) := (D_{21}(\theta)D_{21}'(\theta))^{-1}$; by assumption LPV3, $W(\theta)$ is well-defined for all $\theta \in \Theta$. The closed loop system with states $[x' \;\; x' - \hat{x}']'$ has realization

$$\mathcal{F}_\ell(P(\theta), K(\theta)) = \left[\begin{array}{c|c} \bar{A}(\theta) & \bar{B}(\theta) \\ \hline \bar{C}(\theta) & 0 \end{array}\right]$$

$$:= \left[\begin{array}{cc|c} A(\theta) + B_2(\theta)\bar{J} & -B_2(\theta)\bar{J} & B_1(\theta) \\ 0 & A(\theta) - \alpha Y_\alpha C_2'(\theta)W(\theta)C_2(\theta) & B_1(\theta) - \alpha Y_\alpha C_2'(\theta)W(\theta)D_{21}(\theta) \\ \hline C_1(\theta) + D_{12}(\theta)\bar{J} & -D_{12}(\theta)\bar{J} & 0 \end{array}\right]$$

By corollary 3.1, $N_\alpha(\mathcal{F}_\ell(P(\theta), K(\theta))) < \gamma$ for all $\theta \in \mathcal{F}_\Theta$ if and only if

$$\begin{aligned} \bar{A}(\theta)Q + Q\bar{A}'(\theta) + \alpha Q + \frac{1}{\alpha}\bar{B}(\theta)\bar{B}'(\theta) &\leq 0 \\ \bar{C}'(\theta)\bar{C}(\theta) &< \gamma^2 Q^{-1} \end{aligned}$$

Now, suppose that a matrix Q is set to

$$Q = \begin{bmatrix} Z & Y_\alpha \\ Y_\alpha & Y_\alpha \end{bmatrix},$$

Since $Z > 0$ and $Y_\alpha > 0$, $Q > 0$ if and only if $Z > Y_\alpha$, which must be true since $Z - Y_\alpha$ is a principle minor of the matrix in (18).

Now, consider

$$E(\theta) := \begin{bmatrix} E_{11}(\theta) & E_{12}(\theta) \\ E_{12}'(\theta) & E_{22}(\theta) \end{bmatrix} = \bar{A}(\theta)Q + Q\bar{A}'(\theta) + \alpha Q + \frac{1}{\alpha}\bar{B}(\theta)\bar{B}'(\theta)$$

Performing the algebra, the terms of the matrix simplify to

$$\begin{aligned} E_{11}(\theta) &= A(\theta)Z + ZA'(\theta) + B_2(\theta)V + V'B_2'(\theta) + \alpha Z + \frac{1}{\alpha}B_1(\theta)B_1'(\theta) && (21) \\ E_{12}(\theta) &= A(\theta)Y_\alpha + Y_\alpha A'(\theta) - \alpha Y_\alpha C_2'(\theta)W(\theta)C_2(\theta)Y_\alpha + \alpha Y_\alpha + \frac{1}{\alpha}B_1(\theta)B_1'(\theta) \\ &\quad -B_1(\theta)D_{21}(\theta)W(\theta)C_2(\theta)Y_\alpha && (22) \\ E_{22}(\theta) &= A(\theta)Y_\alpha - \alpha Y_\alpha C_2'(\theta)W(\theta)C_2(\theta)Y_\alpha + Y_\alpha A'(\theta) + \alpha Y_\alpha \\ &\quad +\frac{1}{\alpha}[B_1(\theta)B_1'(\theta) - \alpha B_1(\theta)D_{21}'(\theta)W(\theta)C_2(\theta)Y_\alpha - \alpha Y_\alpha C_2'(\theta)W(\theta)D_{21}(\theta)B_1'(\theta)] && (23) \end{aligned}$$

Under the assumption $B_1(\theta)D_{12}'(\theta) = 0$, $E_{12}(\theta) = E_{22}(\theta)$, so $E(\theta) \leq 0$ for all $\theta \in \Theta$ if $E_{11}(\theta) \leq E_{22}(\theta) \leq 0$.

That $E_{22}(\theta) \leq 0$ for all $\theta \in \Theta$ is immediately evident from (19). From (17), for all $\theta \in \Theta$,

$$A(\theta)(Z - Y_\alpha) + (Z - Y_\alpha)A'(\theta) + B_2(\theta)V + V'B_2'(\theta) + \alpha(Z - Y_\alpha) + \alpha Y_\alpha C_2'(\theta)W(\theta)C_2(\theta)Y_\alpha \leq 0$$

which is equivalent to

$$A(\theta)Z + ZA'(\theta) + B_2(\theta)V + V'B_2'(\theta) + \alpha Z \leq A(\theta)Y_\alpha + Y_\alpha A'(\theta) + \alpha Y_\alpha - \alpha Y_\alpha C_2'(\theta)W(\theta)C_2(\theta)Y_\alpha$$

and

$$\begin{aligned} & A(\theta)Z + ZA'(\theta) + B_2(\theta)V + V'B_2'(\theta) + \alpha Z + \frac{1}{\alpha}B_1(\theta)B_1(\theta)' \\ \leq\ & A(\theta)Y_\alpha + Y_\alpha A'(\theta) + \alpha Y_\alpha - \alpha Y_\alpha C_2'(\theta)W(\theta)C_2(\theta)Y_\alpha + \frac{1}{\alpha}B_1(\theta)B_1(\theta)' \qquad (24) \end{aligned}$$

This expression, in turn, simplifies to

$$E_{11}(\theta) \leq E_{22}(\theta) \ \ \forall\theta \in \Theta$$

thus implying that $E(\theta) \leq 0$, or

$$\bar{A}(\theta)Q + Q\bar{A}'(\theta) + \alpha Q + \frac{1}{\alpha}\bar{B}(\theta)\bar{B}'(\theta) \leq 0 \ \ \forall\theta \in \Theta \qquad (25)$$

Now to demonstrate that $\gamma^2 Q^{-1} > \bar{C}'(\theta)\bar{C}(\theta)$. Through an elementary Schur complement argument, it is easy to show that

$$\gamma^2 Q^{-1} > \bar{C}'(\theta)\bar{C}(\theta) \Leftrightarrow \begin{bmatrix} Q & Q\bar{C}(\theta) \\ \bar{C}'(\theta)Q & \gamma^2 I \end{bmatrix} > 0$$

Substituting for Q and $\bar{C}(\theta)$, and multiplying on both sides by $blockdiag(I, Y_\alpha^{-1}, I)$, this expression is equivalent to

$$\begin{bmatrix} Z & I & ZC_1'(\theta) + V'D_{12}'(\theta) \\ I & Y_\alpha^{-1} & C_1'(\theta) \\ C_1(\theta)Z + D_{12}(\theta)V & C_1(\theta) & \gamma^2 I \end{bmatrix} > 0$$

$$\Leftrightarrow \begin{bmatrix} Z & ZC_1'(\theta) + V'D_{12}'(\theta) & I \\ C_1(\theta)Z + D_{12}(\theta)V & \gamma^2 I & C_1'(\theta) \\ I & C_1(\theta) & Y_\alpha^{-1} \end{bmatrix} > 0 \quad \forall \theta \in \Theta$$

Since $Y_\alpha > 0$, this last expression is equivalent, via Schur complement arguments, to

$$\begin{bmatrix} (Z - Y_\alpha) & (Z - Y_\alpha)C_1'(\theta) + V'D_{12}'(\theta) \\ C_1(\theta)(Z - Y_\alpha) + D_{12}(\theta)V & \gamma^2 I - C_1(\theta)Y_\alpha C_1'(\theta) \end{bmatrix} > 0$$

which is just (18).

To show that the equation is necessary and sufficient when θ is a singleton, θ_0, note that Y_α is now the solution to the ARE

$$(A(\theta_0) + \frac{1}{2}\alpha I)Y_\alpha + Y_\alpha(A(\theta_0) + \frac{1}{2}\alpha I)' - \alpha Y_\alpha C_2'(\theta)W(\theta_0)C_2(\theta_0)Y_\alpha + \frac{1}{\alpha}B_1(\theta_0)B_1'(\theta_0) = 0$$

Then, the matrix inequality in (17), which is equivalent to (24), is now equivalent to

$$A(\theta_0)Z + ZA'(\theta_0) + B_2(\theta_0)V + V'B_2'(\theta_0) + \alpha Z + \frac{1}{\alpha}B_1(\theta_0)B_1(\theta_0)' \leq 0$$

which, by Schur complements, is equivalent to

$$\begin{bmatrix} A(\theta_0)Z + ZA'(\theta_0) + B_2(\theta_0)V + V'B_2'(\theta_0) + \alpha Z & B_1(\theta_0) \\ B_1'(\theta_0) & -\alpha I \end{bmatrix} \leq 0$$

By theorem 3.2, (17) and (18) are necessary and sufficient conditions. □

Remark: The restriction that Y_α in (19) be a minimal stabilizing solution is overly restrictive for the existence of a self-scheduled controller. However, if this restriction is removed, the results do not reduce to necessary and sufficient conditions in the LTI case.

5 Implementation

Theorem 4.1 represents the key analytical device for the synthesis of LPV controllers satisfying prespecified amplitude performance constraints. The approach is identical in nature to that proposed earlier for LTI

systems [1] and for LTI controllers for LPV systems [6]: perform a line search for values of $\alpha > 0$, evaluating (at each α) the conditions of theorem 4.1 to determine if a controller exists satisfying performance constraint, γ. When values of α are found such that the matrix inequality conditions (17)-(18) admit a solution, then the LPV controller with realization given in (20) will achieve closed loop performance goals.

The matrix inequality conditions (17)-(18) of the theorem must be satisfied pointwise for all $\theta \in \Theta$. There are several options to tackle this problem which have been proposed in the literature for similar matrix inequality conditions. When Θ is a polytope, then it is sufficient to evaluate the matrix inequalities simultaneously at the vertices of the polytope. The resulting set of *linear* matrix inequalities can typically be solved quickly when the dimension of the parameter space is small enough. this approach is illustrated in [3]. If, in addition, the parameter dependence in the system matrix coefficients is linear fractional, then the S-procedure[16] may also be applied as in [6]; however, this approach is known to give conservative results[4].

When the parameter set is not polytopic, different approaches must be taken. Perhaps the most obvious approach is to form a grid over the parameter set Θ and simultaneously solve a set of *linear* matrix inequalities, refining the grid until it is apparent that the solution will be valid for points not on the grid. This is the approach taken in [10]. Another approach is to attempt to satisfy the conditions for a polytopic set which contains Θ. The problem here is that the results are potentially more conservative since they must account for parameter variations which will not occur; worse, controllability or observability may be lost at some points in the larger parameter set.

In summary, there are a variety of methods to handle the parameter variations of the matrix inequality conditions (17)-(18). The most attractive require simple parameter set geometry and linear fractional parameter dependence. In other cases, more *ad hoc* approaches must be adopted.

6 Conclusions

This paper has presented sufficient conditions for the synthesis of parameter-dependent output feedback controllers which guarantee an upper bound on the induced-$\mathcal{L}_\infty$ norm of the controlled system. These conditions reduce to necessary and sufficient conditions when the parameter variation is removed, i.e. when plant is LTI. The condition takes the form of linear matrix inequalities (LMIs) which must be solved pointwise over a parameter set. A set of LPV controllers which satisfy the performance constraints are parameterized in terms of feasible solutions to the LMI existence conditions.

The results extend the class of performance problems which may be treated with self-scheduled control

techniques; current literature appears to contain results for the $\mathcal{L}_2$ performance problem only. Furthermore, the results presented here provide a means of bounded amplitude control of class of continuous-time systems more general than previously considered, without the problem of controller dimensionality explosion. They are intuitively satisfying because they are similar in flavor to the $\mathcal{H}_\infty$ results of [5]; parameter-dependent controller synthesis requires the feasibility of a 'controller' Riccati inequality (17), an 'observer' Riccati inequality (19) and a 'spectral radius' coupling inequality (18). Thus, the approach proposed here for bounded amplitude control is no more difficult than similar approaches, e.g. $\mathcal{L}_2$ control, though they may be more computationally intensive. However, note that the results presented here do not rely on the loop-shifting arguments used in, e.g. [12],[2], so that well-posedness issues in the controller implementation are not an issue on the results presented here.

Finally, it is apparent that the conditions presented here might be strengthened to necessary and sufficient if the Riccati matrices Y_{min}, Z and V of the theorem are allowed to be parameter dependent. In that case, specific assumptions must be imposed in the rate of parameter variation, as in the $\mathcal{L}_2$ case.

References

[1] John Abedor, Krishan Nagpal, and Kameshwar Poolla. A linear matrix inequality approach to peak-to-peak gain minimization: Filtering and control. Technical report, University of California, Berkeley, 1994.

[2] P. Apkarian and P. Gahinet. A convex characterization of gain-scheduled H_∞ controllers. *IEEE Transactions on Automatic Control*, 40(5):853–864, May 1995.

[3] P. Apkarian, P. Gahinet, and G. Becker. Self-scheduled H_∞ control of linear parameter-varying systems: a design example. *Automatica*, 31(9):1251–1261, September 1995.

[4] Pierre Apkarian and Richard J. Adams. Advanced gain-scheduling techniques for uncertain systems. In *Proceedings of the American Control Conference*, Albuquerque, NM, June 1997.

[5] G. Becker and A. Packard. Robust performance of linear parametrically varying systems using parametrically-dependent linear feedback. *Systems and Control Letters*, 23:205–215, 1994.

[6] Christopher J. Bett and Michael Lemmon. Finite-horizon induced-L_∞ performance of linear parameter varying systems. In *Proceedings of the American Control Conference*, Albuquerque, NM, June 1997.

[7] Stephen Boyd, Laurent El Gaoui, Eric Feron, and Venkataramanan Balakrishnan. *Linear Matrix Inequalities in System and Control Theory.* Society for Industrial and Applied Mathematics, 1994.

[8] M.A. Dahleh and I.J. Diaz-Bobillo. *Control of uncertain systems: a linear programming approach.* Prentice-Hall, 1995.

[9] M.A. Dahleh and J.B. Pearson. L^1-optimal compensators for continuous-time systems. *IEEE Transactions on Automatic Control*, 32(10):889–895, October 1987.

[10] Edward W. Kamen and Pramod P. Khargonekar. On the control of linear systems whose coefficients are functions of parameters. *IEEE Transactions on Automatic Control*, AC-29(1):25–33, January 1984.

[11] K. Nagpal, J. Abedor, and K. Poolla. An LMI approach to peak-to-peak gain minimization: filtering and control. In *Proceedings of the American Control Conference*, pages 742–746, Baltimore, Maryland, June 1994.

[12] Andy Packard. Gain scheduling via linear fractional transformations. *Systems and Control Letters*, 22:79–92, 1994.

[13] Jeff S. Shamma and Michael Athans. Analysis of gain scheduled control for nonlinear plants. *IEEE Transactions on Automatic Control*, 35(8):898–907, 1990.

[14] Jeff S. Shamma and Michael Athans. Guaranteed properties of gain scheduled control for linear parameter-varying plants. *Automatica*, 27(3):559–564, 1991.

[15] M. Vidyasagar. Optimal rejection of persistent bounded disturbances. *IEEE Transactions on Automatic Control*, 31(6):527–535, June 1986.

[16] V.A. Yakubovich. The S-procedure in nonlinear control theory. *Vestnik Leningrad University Mathematics*, 4:73–93, 1977.

[17] Kemin Zhou, John C. Doyle, and Keith Glover. *Robust and Optimal Control.* Prentice-Hall, Inc., Upper Saddle River, New Jersey, 1996.

APPENDIX F

K.X. He and M.D. Lemmon
"Lyapunov Stability of Continuous-valued Systems under the Supervision of Discrete-Event Transition Systems"
to appear in Hybrid Systems: computation and control,
International Workshop, Berkeley California, USA, April 13-15, 1998.
Also released as ISIS Technical Report ISIS-97-010, September 1997,

Lyapunov Stability of Continuous-Valued Systems Under the Supervision of Discrete-Event Transition Systems

Kevin X. He and Michael D. Lemmon
Dept. of Electrical Engineering
University of Notre Dame
Notre Dame, IN 46556, USA
(219)-631-8309
fax:(219)-631-4393
lemmon@maddog.ee.nd.edu

1 Introduction

This paper examines the Lyapunov stability of equilibrium points for switched control systems [Morse95]. A switched control system is a continuous-valued system whose control law is switched in a discontinuous manner as the system state evolves over a continuous-valued subset of $\Re^n$. Of particular interest in this paper are switched systems in which the switching logic is generated by a discrete-event transition system that can be represented as either a finite automaton or bounded Petri net.

There are a variety of prior results identifying sufficient conditions for such systems to be Lyapunov stability. In [Pel91] and [Sav96] a single positive definite functional is found which is Lyapunov for all control systems in the collection. Multiple Lyapunov function approaches in [Bran94] and [Hou96] have been presented which should be applicable to a larger set of systems than the single Lyapunov function methods. In certain cases, where the switched systems are linear time invariant and the switching regions are defined by conic sectors, it has been suggested that candidate Lyapunov functionals can be numerically computed by finding feasible points of a linear matrix inequality [Pet96] [Rant97].

While these prior results have provided great insight into the Lyapunov stability of switched systems, they do not account for the actual switching laws used by the system. In the case of the computational methods proposed in [Pet96] and [Rant97] this leads to LMI's which are extremely large and hence provide an overly restrictive sufficient condition for switched system stability. This paper examines the numerical question and asks what sort of information about the switching law can be used to significantly reduce the computational complexity and conservatism associated with finding candidate Lyapunov functions of switched systems. The principal result of this paper states that if the switching law can be represented as a discrete-event transition system such as a finite automaton or Petri net, then it suffices to examine the fundamental cycles of the directed graph associated with such structures to assess switched system stability. In particular, the results and viewpoints suggested in this paper provide a way in which the traditional control theoretic methods cited above can be combined with results from computer science [Alur94] [Alur96] concerned with the behaviour of timed transition systems.

The remainder of the paper is organized as follows. In section 2, we first introduce a formal model for switched control systems which are supervised by a discrete-event transition system. Section 3 states recent results [Bran94] [Pet96] providing sufficient conditions for switched system stability using a multiple Lyapunov function approach. Section 4 motivates, states, and proves the paper's principal result. Section 5 presents an example illustrating the value of using fundamental cycles in assessing switched system stability. Section 6 concludes with topics and directions for further study.

2 Problem Statement

Let $X \subset \Re^n$ be a smooth n-dimensional manifold and let I be a finite set of N integers. Let Δ be a constant dimensional distribution,

$$\Delta = \{f_1, \ldots, f_N\} \tag{1}$$

where $f_i : X \to X$ for $i = 1, \ldots, N$ are locally Lipschitz vectorfields over X. We consider **switched dynamical system** to be described by the following set of equations.

$$\dot{x}(t) = f_{i(t)}(x(t)) \tag{2}$$

$$i(t) = q(x(t), i(t^-)) \tag{3}$$

where $x : \Re \to X$, $i : \Re \to I$, and $q : X \times I \to I$. $i(t^-)$ refers to the lefthand limit of the function $i(t)$ at point t. In the sequel, we refer to each f_i as a **subsystem** of the switched system. The preceding model is essentially that used in [Tav87].

A **trajectory** of the switched system is the ordered pair, (x, i), where $x : \Re \to X$ and $i : \Re \to I$ which **solves** the system equation. The value taken by the trajectory at time $t \in \Re$ is denoted by $(x(t), i(t))$. We say that (x, i) **solves** the system equation if and only if the equations are satisfied by $x(t)$ and $i(t)$ for all $t \in \Re$. This paper does not treat questions concerned with the existence of solutions. In general, however, solutions (when they do exist) will not be unique due to the nondeterminism in the switching law.

Let (x, i) be the trajectory generated by a switched dynamical system. The set of **switching times**, Ω, of a trajectory (x, i) will be

$$\Omega = \left\{ t : \lim_{\tau \to t^+} i(\tau) \neq \lim_{\tau \to t^-} i(\tau) \right\} \tag{4}$$

The set of **switching events**, $\mathcal{E}$, of trajectory (x, i) is denoted as

$$\mathcal{E} = \left\{ (i, t) \in I \times \Re : t \in \Omega, i = \lim_{\tau \to t^+} i(\tau) \right\} \tag{5}$$

We define the **timed projection** $P_t : \mathcal{E} \to \Re$ by the equation $P_t[(i, \tau)] = \tau$ and the **event projection**, $P_e : \mathcal{E} \to I$ by the equation $P_e[(j, \tau)] = j$.

The **switching sequence**, is a mapping $\lambda : Z \to \mathcal{E}$ such that

$$P_t[\lambda(n)] < P_t[\lambda(n+1)] \tag{6}$$

for all $n \in Z$. Suppose λ is a switching sequence. Let I^* be the set of all strings formed from I. We let $\lambda_e = P_e[\lambda] \in I^*$ and $\lambda_t = P_t[\lambda]$ denote the event and time projections of λ, respectively. Let the subsequence of times when system j is turned on and off be denoted as $\lambda_{t,j} \in I^*$. In other words,

$$\lambda_{t,j} = \lambda_t(n_1), \lambda_t(n_1+1), \cdots \lambda_t(n_k), \lambda_t(n_k+1), \cdots \tag{7}$$

where n_k is a subsequence of Z such that $P_e[\lambda(n_k)] = j$. Define the interval completion $I(\lambda_{t,j})$ as the set obtained by taking the union of all open intervals in which system j is active. In other words,

$$I(\lambda_{t,j}) = \bigcup_{k=1}^{\infty} (\lambda_t(n_k), \lambda_t(n_k+1)) \tag{8}$$

Denote $E(\lambda_{t,j})$ as a subsequence of $\lambda_{t,j}$ when the subsystem j is turned on. In other words,

$$E(\lambda_{t,j}) = \lambda_t(n_1), \lambda_t(n_2), \cdots, \lambda_t(n_k), \cdots \tag{9}$$

The preceding model of a switched system assumes a very general switching function, q. To obtain more precise results, however, we need to specify the nature of the switching function. A common choice is to associate a **discrete-event** transition system such as a finite automaton or Petri net with the switching system. In this paper we limit our scope to finite automata. An automaton is tied to the switched system by associating the vertices to the switched system's subsystems and by associating the arcs with *switching sets* called *guards*. The timed automaton [Alur94] and hybrid automaton [Alur96] provide tangible examples of this approach. In this paper we begin by considering a discrete-event transition system that is represented by a finite automaton, (V, A).

A **finite automaton** associated with the switched system is the directed graph (V, A) where $V = I$ is a set of vertices and $A \subset V \times V$ is a set of directed arcs. By definition, the automaton associates a subsystem f_i with each vertex of the (V, A). We define the **guard**, Ω_{ij} of arc $(i, j) \in A$ as

$$\Omega_{ij} = \{x \in X : j = q(x, i)\} \tag{10}$$

The ordered pair (i, j) is an arc of A if and only if $\Omega_{ij} \neq \emptyset$. The guard therefore represents a subset of the switched system's state space in which a switch can occur. The guard set Ω_{ii} will sometimes be denoted as Ω_i and represents the set in which subsystem f_i remains active.

The preceding paragraph characterized the switching logic by a finite automaton (V, A). It is straightforward to generalize this approach to consider more complex switching logics. In particular, let's consider how this might be done for a switching logic generated by a Petri net. A Petri net is represented by a directed graph (V, A) where the vertex set consists of two types of vertices, **places**, P, and **transitions**, T. The vertex set, therefore, takes the form

$P \times T = V$. We associate this directed graph structure with the switched system by letting $P = I$. We therefore associate a subsystem with each place of the Petri net. The guards, Ω_{ij}, are associated with the transition $t \in T$ which connect the ith and jth places of the network. Petri nets provide natural structures for modeling concurrency and synchronization in parallel systems. In general, a Petri net can provide a more expressive characterization of a system's switching logic than can be provided by a finite automaton.

Let (x, i) be the trajectory generated by a switched dynamical system. The trajectory is said to be **deadlock free** if the event projection of the switching sequence $P_e[\lambda]$ is not finite. We say that the trajectory is **live** if the event projection of the switching sequence $P_e[\lambda]$ contains an infinite number of each index, $i \in I$. In other words any subsystem can be switched an infinite number of times in a switching sequence. We say that the trajectory is **nonZeno** if the timed projection of the switching sequence $P_t[\lambda]$ satisfies

$$\sum_{n=1}^{\infty} P_t[\lambda(n)] > \infty \tag{11}$$

We say that the switched system is live, deadlock free, or nonZeno if all of its trajectories are live, deadlock free, or nonZeno, respectively.

An important issue which is not addressed in this paper concerns necessary and sufficient conditions for a switched system to be live, deadlock free, or nonZeno. In this paper, we assume that the switched system is live and nonZeno.

3 Prior Results

This section briefly discusses prior results on switched system stablity. Let (x, i) be any trajectory generated by the switched dynamical system. Assume that $f_i(0) = 0$ for all $f_i \in \Delta$. The equilibrium point $x = 0$ is said to be **stable in the sense of Lyapunov** if and only if for all $\epsilon > 0$ there exists $\delta > 0$ such that $\|x(t_0)\| < \delta$ implies $\|x(t)\| < \epsilon$ for all $t \geq t_0$.

In the following we will denote the open ball of radius r centered at the origin as

$$B(r) = \{x \in \Re^n \ : \ \|x\| < r\} \tag{12}$$

The sphere, $S(r)$, of radius r centered at the origin is the set

$$S(r) = \{x \in \Re^n \ : \ \|x\| = r\} \tag{13}$$

Let λ be a switching sequence for a switched dynamical system where λ_t is its time projection. we say that a continuously differentiable function $V : \Re^n \to \Re^+$ is **Lyapunov-like function** over sequence λ_t if and only if $\dot{V}(x(t)) \leq 0$ for all $t \in \mathcal{I}(\lambda_t)$ and V is monotonically nonincreasing on $E(\lambda_t)$. Using this definition of a Lyapunov like function, the following sufficient condition for Lyapunov stability was proven in [Bran94]. The proof uses standard techniques employed in proving Lyapunov stability for nonautonomous systems. A significant generalization of this result will be found in [Hou96].

Theorem 1. *Suppose we have candidate Lyapunov functions V_j ($j \in I$) and suppose that the switched system is nonZeno and satisfies $f_i(0) = 0$ for all $j \in I$. If V_j is a Lyapunov like function for switching sequence $\lambda_{i,j}$ for all $j \in I$. then the equilibrium point $x = 0$ of the switched system is stable in the sense of Lyapunov.*

The preceding theorem provides a sufficient condition for Lyapunov stability of switched systems. The condition requires that a set of Lyapunov like functions be determined for all possible switching sequences λ that can be generated by the system. The determination of Lyapunov like functions may not be possible in general. For switched systems in which each subsystem is a linear time invariant system and the guard sets are represented by conic sectors in $\Re^n$, a method for determining the Lyapunov like functions was presented in [Pet96] and [Rant97]. Assume that each subsystem can be written as

$$\dot{x}(t) = A_i x(t) \tag{14}$$

where $A_i \in \Re^{n \times n}$ and $i \in I$. Assume that the guard sets can be bounded by conic sectors parameterized by symmetric matrices Q_{ij}. In other words, consider sets,

$$\Omega_{ij} \subseteq \{x \in \Re^n | x' Q_{ij} x \leq 0\} \tag{15}$$

Ω_{ii} represents the set in which the ith subsystem is free to operate and Ω_{ij} (where $i \neq j$) denotes the guard set for the transition between the ith and jth vertices. If we can find real matrices, $P_i = P_i' > 0$ for all $i \in I$ and real constants $\alpha_i > 0$ and $\alpha_{ij} > 0$ such that

$$A_i' P_i + P_i A_i + \alpha_i Q_{ii} \leq 0 \tag{16}$$

$$P_i - P_j + \alpha_{ij} Q_{ij} \leq 0 \ , \tag{17}$$

then the functionals, $V_j = x' P_j x$ are Lyapunov like functions of the switched system. This particular conditions is more restrictive than that formulated in [Bran94]. But it can be readily reformulated as a linear matrix inequality (LMI) which can be solved using interior-point methods for convex optimization.

4 Main Result

The sufficient conditions presented in [Bran94] [Hou96] and used in [Pet96] [Rant97] to compute candidate Lyapunov functionals provide a very conservative approach for testing switched system stability. In the first place, the stability theorems in [Bran94] [Hou96] require that V_j be Lyapunov like for all possible switching sequences. These papers place no assumptions on the nature of the switching laws used so that when the computational methods of [Pet96] and [Rant97] are employed, the worst case switching law has to be considered. The worst case switching law is one in which every switch is possible. This assumption can result in an extremely high dimensional linear matrix inequality which may be more restrictive than it needs to be.

In this section, we present and prove a result which shows that when the switching logic can be characterized by a finite automaton, then we only need to search for Lyapunov like functions over a restricted set of **fundamental cycles** in the finite automaton. Essentially, the following result shows that rather than having to examine whether a set of candidate functions are Lyapunov like for all possible switching sequences, we only need consider whether the candidate functions are Lyapunov like over a potentially smaller sized set of fundamental cycles. In section 5, this result is used to significantly reduce the computational complexity and conservatism of the LMI method suggested in [Pet96] and [Rant97].

Let the directed graph (V, A) have $n+1$ vertices, $i_0, i_1, \cdots, i_n$. The sequence of arcs

$$(i_0, i_1), (i_1, i_2), \cdots, (i_{n-1}, i_n) \tag{18}$$

is called a path of length n. A **cycle** of a directed graph is any path such that $i_0 = i_n$. A cycle of length n

$$(i_0, i_1), (i_1, i_2), \cdots, (i_{n-1}, i_0) \tag{19}$$

is said to be **fundamental** if $i_j \neq i_k$ for all j, k not equal to zero or n and for all $j \neq k$. The following results are basic facts from graph theory. In any fundamental cycle, any two vertices are connected by one and only one path. An arc of a directed graph that is in a cycle is also in a fundamental cycle. For any cycle, C, in a directed graph, there exists a set of fundamental cycles $C_1, C_2, \cdots, C_N$ such that

$$\text{Arcs}(C) = \bigcup_{i}^{N} \text{Arcs}(C_i) \tag{20}$$

Finally, the fundamental cycles of a directed graph can be determined in polynomial time by constructing a minimal spanning tree for the graph.

To state and prove the main result of this paper, we first need to establish some facts about fundamental cycles generated by live switched systems. The first principal lemma is a result saying that any event sequence generated by a switched system can be constructed by recursively inserting fundamental cycles into a legal switching sequence. We then introduce a sufficient condition for a fundamental cycle to be **uniformly bounded** with respect to time. These two results are then combined to establish the Lyapunov stability of the entire switched system.

Lemma 2. *In the automaton associated with a live switched system, every arc is in at least one fundamental cycle*

Proof: Let (V, A) denote the finite automaton associated with a switched system. Assume that there exists an arc $(i, j) \in A$ which is not in any cycle of (V, A). Therefore, once we go through arc (i, j) then there is no path back to vertex $i \in V$. Therefore in any switching sequence λ that contains arc (i, j)

the number of times when vertex i is reached will be reached is finite which contradicts the definition of a live transition system. Therefore every arc of a live automaton is in a cycle. Furthermore from the fundamental results about cycles in directed graphs, we know that every arc is in at least one fundamental cycle, so the the lemma is proven. •

Lemma 3. *Any switching sequence λ generated by a live switched system can be decomposed as*

$$\lambda_e = \sigma_1\sigma_2\sigma_3 \tag{21}$$

where σ_1 is a prefix of λ_e, σ_3 is a suffix of λ_e, and σ_2 is a fundamental cycle of the switched system's automaton.

Proof: Assuming there exists a switching sequence λ with event projection λ_e such that the decomposition doesn't exist. This means that there is no substring in λ_e which is a fundamental cycle. But from the definition of a live switched system, we know that every arc must be in a cycle. Let i_1 be the vertex where such a cycle starts. If the cycle is fundamental, then we have a contradiction and the proof is finished. But if the cycle is not fundamental, then there is a vertex i_2 which is crossed more than once in the cycle. Consider the cycle starting from i_2. Either this cycle is fundamental, or not. If not, then we can repeat the above argument to find a smaller cycle within this one. However, because the automaton is finite, this recursion has to terminate in a fundamental cycle. We therefore have a contradiction and the lemma is proven. •

Proposition 4. *Given a switching sequence λ generated by a live switched system, let $\Lambda : Z \to I^*$ be a sequence of sequences in I^* constructed by the recursive procedure:*

1. *$\Lambda[0]$ is a fundamental cycle C_0*
2. *$\Lambda[n] = \sigma_1 C_n \sigma_2$ where $\sigma_1\sigma_2 = \Lambda[n-1]$ and C_n is a fundamental cycle.*

Then there exists a set of C_i such that $\Lambda[n]$ is a prefix of λ for all n.

Proof: From lemma 3 we know that any switching sequence can be decomposed to $\sigma_1\sigma_2\sigma_3$ where σ_2 is a fundamental cycle. Note that if we pull out σ_2 from the switching sequence, then $\sigma_1\sigma_3$ is still a possible switching sequence. We can now decompose the resulting sequence $\sigma_1\sigma_3$ using lemma 3 to pull out another fundamental cycle of the automaton. Since the switching sequence is countable, we can repeat this process to pull out a countable sequence of fundamental cycles. This sequence is the set of C_i referred to in the above proposition. •

A given sequence of events can be generated in various ways by a switched system. What we'd like to do is ensure that the cycle is well-behaved in some appropriate sense. In particular, we'll require that the continuous-state trajectory over the cycle is uniformly bounded with respect to time. The following lemma provides sufficient conditions for the system to be uniformly bounded.

Lemma 5. *Let λ_e be any cycle generated by the live switched system consisting of events*

$$\lambda_e = j_1, \ldots, j_K \tag{22}$$

where $j_{K+1} = j_1$ with switching times

$$t_0, t_1, \cdots t_K \tag{23}$$

So that t_i is the time when the ith system is switched off and the $i+1$st system is switched on.

If there exist a set of continuously differentiable functions $V_j : \Re^n \to \Re$ for $j \in I$ such that $\dot{V}_j(x(t)) \leq 0$ for all $t \in [t_{j-1}, t_j)$, then for any $\epsilon > 0$ there $\delta(\epsilon) > 0$ such that for all $\|x(t_0)\| < \delta(\epsilon)$, $\|x(t)\| < \epsilon$ for all $t \in [t_0, t_K]$.

Proof: Consider an arbitrary $\epsilon > 0$ and let

$$\beta_K = \min_{x \in S(\epsilon)} V_{j_K}(x) \tag{24}$$

Define the closed set,

$$\Omega_K = \{x \in B(\epsilon) : V_{j_K}(x) \leq \beta_K\} \tag{25}$$

Choose ρ_K such that for all $x \in B(\rho_K)$, we have $V_{j_K}(x) < \beta_K$. We now define

$$\beta_{K-1} = \min_{x \in S(\rho_K)} V_{j_{K-1}}(x) \tag{26}$$

and introduce the closed set,

$$\Omega_{K-1} = \left\{x \in B(\rho_K) : V_{j_{K-1}}(x) \leq \beta_{K-1}\right\} \tag{27}$$

Choose ρ_{K-1} as was stated above and continue this process to construct a monotone sequence of sets

$$\Omega_1 \subseteq \Omega_2 \subseteq \cdots \subseteq \Omega_{K-1} \subseteq \Omega_K \tag{28}$$

Note that Ω_j is invariant with respect to subsystem f_j because of the condition on $\dot{V}_j$. Therefore, we expect that if we start in $B(\rho_0)$, we should stay in set $B(\epsilon)$, which is sufficient to establish the lemma's conclusion. •

A cycle for which such functionals can be found will be said to be **uniformly bounded**. We now state and prove the main result of this section. This result uses the preceding proposition to show by induction that each of the sequences in the supersequence of lemma 3 is uniformly bounded if each fundamental cycle is uniformly bounded.

Theorem 6. *Consider a live nonZeno switched system where $f_j(0) = 0$ for all $j \in I$. Let λ be a switching sequence generated by the system. Let μ denote a subsequence of contiguous switches in λ such that $P_e[\mu]$ is a fundamental cycle of the system's automaton. Let $\bar{\mu}$ denote the infinite sequence formed by concatenation of μ with itself.*

If there exist a set of continuously differentiable functions $V_j : \Re^n \to \Re$ which are Lyapunov like over sequence $\bar{\mu}_{t,j}$ for all $j \in I$, then the system is stable in the sense of Lyapunov.

Proof: From our earlier lemma, we know that any switching sequence can be constructed by inserting fundamental cycles into a legal switching sequence. Let

$$\Lambda = \lambda[0], \lambda[1], \cdots \lambda[n], \cdots \tag{29}$$

By definition $\lambda[0]$ is a fundamental cycle and under the theorem's hypothesis this is uniformly bounded.

Now assume that the sequence $\Lambda[n]$ is uniformly bounded. By assumption the fundamental cycle inserted into $\Lambda[n]$ is uniformly bounded. Note also, however, that since V_j is Lyapunov like we require that if $x(t_0) \in \Omega_1$, then it must return to that set. Hence the addition of the fundamental cycle does not change the boundedness of the original sequence $\Lambda[n]$. We can therefore conclude that $\Lambda[n+1]$ is uniformly bounded.

We now consider the limit as $n \to \infty$. Since the δ determined for uniform boundedness is indepedent of time, we can conclude that it holds for sequences of arbitrary length and hence the system is stable in the sense of Lyapunov.•

5 Example

In this section, we present some examples illustrating the application of the result in the preceding section to the computation of Lyapunov-like functionals using the LMI methods of [Pet96] and [Rant97].

Consider a live switched system whose automaton is shown in figure 1. Associated with each vertex is an LTI subsystem of the form

$$\dot{x} = A_i x \tag{30}$$

where $i = 1, 2, \ldots, 6$. In addition to $A_i \in \Re^{2\times 2}$, we associate the "self-switching" set characterized by the symmetric matrix Q_i. Figure 1 shows the given automaton and the assumed matrices associated with each vertex. Each arc (i, j) in the automaton has a matrix Q_{ij} associated with it. The arcs are shown in figure 1 also.

From the automaton we can identify a set of three fundamental cycles. These fundamental cycles are obtained by determining a minimal spanning tree for the automaton's directed graph. This directed graph is shown in figure 2 and the resulting fundamental cycles are $1-2-3$, $1-4-3$ and $5-6-2-3$, respectively.

From the theorem proven above, we know that it suffices to find a set of continuously differentiable functions, V_j, which are Lyapunov-like for each fundamental cycle in the automaton. Determining such Lyapunov-like functions can now be done using the method suggested in [Pet96] and [Rant97]. We establish three sets of matrix inequalities corresponding to the three fundamental cycles. For cycle $1-2-3$, we have the set of inequalities,

$$A_i' P_i + P_i A_i + \alpha_i Q_i \leq 0 \; i = 1, 2, 3$$

$$P_2 - P_1 + \alpha_{12} Q_{12} \leq 0$$

$$P_3 - P_2 + \alpha_{23} Q_{23} \leq 0$$

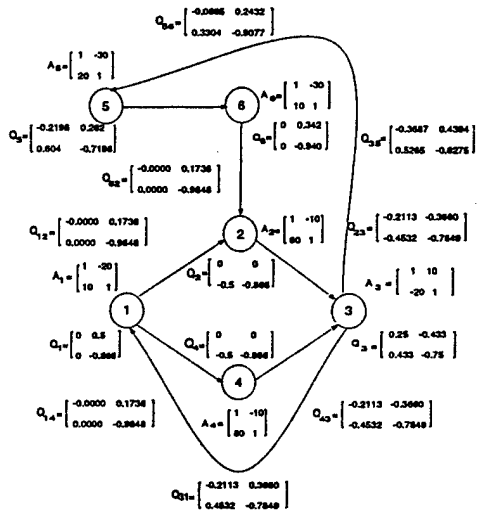


Fig. 1. The automaton of the example live switched system

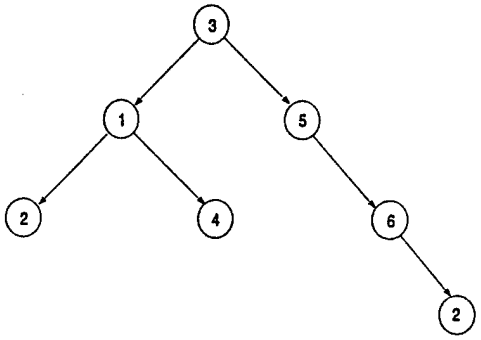


Fig. 2. Spanning Tree Identifying Switched System's Fundamental Cycles

$$P_1 - P_3 + \alpha_{31} Q_{31} \leq 0$$

A similar set of inequalities can be formed for the other three cycles. To find the Lyapunov like functions, $V_i = x' P_i x$, we want to make sure that all fundamental cycles are stable, so we build a large LMI which includes all the matrix inequalities associated with the three fundamental cycles. For this example, there are a total of 14 matrix equations.

The 14 equation LMI is still a large LMI, but it can be readily solved using the LMI toolbox. If we had proceeded using the technique originally proposed in [Pet96], then we would need to build an LMI which accounted for all individual transitions that could possibly happen. If the automaton had N vertices, then we would have N^2 equations in our linear matrix inequality. On the otherhand, if the automaton had M fundamental cycles, then the total size of the LMI would be bounded by $(M+1)N$ since each cycle can have no more than N vertices.

In many cases, this bound is much larger than could be seen. For our particular example, we would have a 36 equation LMI to solve.

The implication of increasing LMI size is that it represents an overly restrictive sufficient condition for system stability. In our case, we can see this quite easily by solving the 14 equation LMI obtained by examining the fundamental cycles of the system versus the 36 equation LMI obtained by using the methods in [Pet96]. The P matrices obtained in both cases for our example are shown in figure 3

	simplified method	original method
P_1	$\begin{bmatrix} 38.3443 & -5.7616 \\ -5.7616 & 65.2333 \end{bmatrix}$	$\begin{bmatrix} 0.0869 & -0.0134 \\ -0.0134 & 0.1470 \end{bmatrix}$
P_2	$\begin{bmatrix} 27.6904 & -1.7744 \\ -1.7744 & 5.7280 \end{bmatrix}$	$\begin{bmatrix} 0.0610 & -0.0042 \\ -0.0042 & 0.0128 \end{bmatrix}$
P_3	$\begin{bmatrix} 35.6272 & 16.0261 \\ 16.0261 & 21.2592 \end{bmatrix}$	$\begin{bmatrix} 0.0761 & 0.0276 \\ 0.0276 & 0.0436 \end{bmatrix}$
P_4	$\begin{bmatrix} 31.3473 & -1.3288 \\ -1.3288 & 4.9189 \end{bmatrix}$	$\begin{bmatrix} 0.0707 & -0.0032 \\ -0.0032 & 0.0112 \end{bmatrix}$
P_5	$\begin{bmatrix} 40.3580 & -3.3383 \\ -3.3383 & 46.6225 \end{bmatrix}$	$\begin{bmatrix} 0.0799 & -0.0075 \\ -0.0075 & 0.0894 \end{bmatrix}$
P_6	$\begin{bmatrix} 36.0113 & -14.3852 \\ -14.3852 & 87.6743 \end{bmatrix}$	$\begin{bmatrix} 0.0740 & -0.0288 \\ -0.0288 & 0.1771 \end{bmatrix}$

(31)

Fig. 3. P matrices for example

The existence of these P matrices indicates that the given system is stable. A simulation of the example system's trajectory is shown in figure 4. This trajectory is clearly stable.

In computing the first table, the LMI toolbox required 33478 flops to determine the P matrices for the original method. The simplified method developed in this paper only required a total of 13310 flops. So our method clearly has a lower computational complexity than the original method of [Pet96]. More important than this, however, is the difference between the matrices. As can be clearly seen above, the singular values for the P matrices obtained from the simplified approach are around 50. For the original approach in [Pet96], however, these values are about .1. Since the singular value is a measure of how close the matrix is to being singular, this means that the original method was almost unable to determine the candidate Lyapunov functions. With minor changes in the Q matrices it is quite possible to generate examples in which the original method is unable to find the required P matrices, but our method would find such matrices.

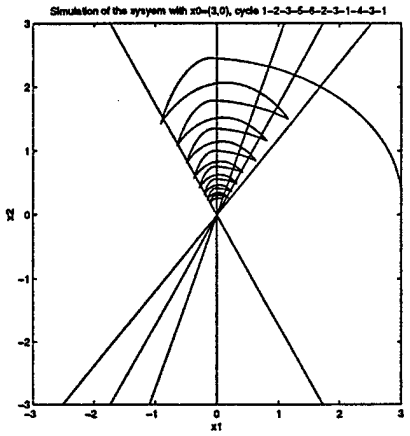


Fig. 4. The trajectory of the sample system under switching cycle

6 Future Work

This paper has presented a sufficient method for switched system stability which takes advantage of prior knowledge of the system's switching logic. In particular, it was shown that if the switching logic can be shown to be generated by a finite discrete-event transitions system such as a finite automaton or Petri net, then it suffices to determine Lyapunov-like functions only over the fundamental cycles of the state machine. This observation can greatly reduce the computational complexity involved in testing for switched system stability as well as providing a less conservative set of Lyapunov functions. These conclusions were validated by using them to compute the Lyapunov functions for an LTI switched system using the LMI method suggested in [Pet96] and [Rant97].

The preliminary results presented in this paper are encouraging and suggest several possible directions for future study. One future direction involves extending the concepts introduced here to study switching logics generated by Petri nets. The use of unfolding methods should allow the efficient identification of fundamental cycles in the Petri net's reachability tree, thereby providing a sufficient test for the stability of such systems. Another promising avenue of future study involves developing sufficient tests for uniform ultimate boundedness (bounded-amplitude) in switched systems. For important classes of systems, we can also formulate these sufficient conditions as matrix inequalities thereby allowing the efficient testing of switched system performance with respect to a specified ultimate bound.

7 Acknowledgements

We gratefully acknowledge the partial financial support of the Army Research Office (DAAH04-95-1-0600) and the National Science Foundation (NSF-ECS95-

31485).

References

[Morse95] A.S. Morse, Control using logic based switching, In A. Isidori, Trends in Control, Springer-Verlag, Great Britain, 1995.

[Pel91] P. Peleties and R. DeCarlo, Asymptotic Stability of m-switched systems using Lyapunov-like functins. In *Proceedings of the American Control Conference*, pp 1679-1684, Bost, MA, June 1991.

[Sav96] A.V. Savkin, I.R. Petersen, E. Skafidas, and R.J. Evans, Robust control via controlled switching, In *Proceedings of CESA'96*, pages 1117-1122, Lille France, 1996

[Bran94] M. Branicky, Stability of Switched and Hybrid Systems, In *Proceedings of the 33rd Conference on Decision and Control*, pp 3498-3503, Lake Buena Vista, FL, December 1994.

[Hou96] L. Hou, A.N. Michel, and H. Ye, Stability Analysis of Switched Systems, In *Proceedings of the 35th Conference on Decision and Control*, pages 1208-1212, Kobe Japan, December 1996.

[Pet96] S. Pettersson and B. Lennartson, Stability and Robustness for Hybrid Systems, In *Proceedings of the 35th Conference on Decision and Control*, Kobe Japan, December 1996.

[Rant97] M. Johansson and A. Rantzer, Computation of Piecewise Quadratic Lyapunov Functions for Hybrid Systems, to appear in *IEEE Trans. of Automatic Control*, 1997

[Alur94] R. Alur and D.L. Dill, A Theory of Timed Automata, *Theoretical Computer Science*, 126:183-235, 1994

[Alur96] R. Alur, T. Henzinger, and P-H Ho., Automatic Symbolic Verification of Embedded Systems, *IEEE Transactions on Software Engineering*, 22:181-201, 1996.

[Tav87] L. Tavernini, Differential automata and their discrete simulators, *Nonlinear Analysis, theory, methods, and applications*, 11(6):665-683, 1987.

[Bran95] . S. Branicky, Studies in Hybrid Systems: Modeling, Analysis, and Control, Technical Report LIDS-TH-2304 of the Laboratory for Information and Decision Sciences, MIT, 1995.

[Desh95] A. Deshpande and P. Varaiya, Viable Control of Hybrid Systems, *Hybrid Systems II*, A. Nerode (ed.), LNCS Volume 999, Springer-Verlag, 1995.

This article was processed using the LaTeX macro package with LLNCS style

APPENDIX G

M.D. Lemmon, K.X He, and C.J. Bett
"Modeling Hybrid Control Systems using Programmable Timed Petri Nets"
to appear in L'Automatisation des Processus Mixtes (ADPM'98)
Rheims, France, March 19-20, 1998.

Modeling Hybrid Control Systems using Programmable Timed Petri Nets

MODELISATION DES SYSTEMES DYNAMIQUES HYBRIDES PAR RESEAUX DE PETRI TEMPORISES PROGRAMMABLES

Michael Lemmon, Kevin He, and C.J. Bett
Department of Electrical Engineering
University of Notre Dame
Notre Dame, IN 46556
Tel: +01-219-631-8309, E-mail: lemmon@maddog.ee.nd.edu

Abstract: Hybrid systems are systems which generate a mixture of discrete-event and continuous-valued signals. This paper presents an extension of Alur's hybrid automaton (Alur 96) in which a timed Petri net is used to model a hybrid system. The resulting modeling framework is called a "programmable" timed Petri net (PTPN). PTPN's provide a way of modeling concurrency in complex dynamical systems. In this paper, the PTPN is used to model hybrid systems obtained by switching between collections of linear time-invariant plants whose switching logic is generated by a PTPN. We use unfoldings of the PTPN to develop sufficient tests for the switched system to have Lyapunov stability and uniform ultimate bounded behaviour. The concepts of this method are illustrated on a power system example.

Résumé: Les systèmes dynamique hybrides désignent des catégories de systèmes de type continu et événementiel. Cette communication présente les réseaux de Petri temporisés "programmables" (PTPN) qui constituent une extension des automates hybrides d'Alur. Les PTPN sont bien adaptés à la représentation des systèmes hybrides parallèles à plusiers modes de fonctionnement. Des résultat concernant la stabilité de Lyapunov et la "bornitude" uniforme maximale des systèmes hybrides sont obtenus à partir d'un dépliage des PTPN.

1. INTRODUCTION

This paper focuses on switched hybrid systems where the individual subsystems are linear time invariant plants with bounded additive disturbances and where the switching logic is generated by a timed Petri net which we call the programmable timed Petri net or PTPN. The principal contribution of this paper is a proposed method for analyzing the stability and uniform ultimate boundedness of switched linear systems represented by PTPN. The methods used in this paper are extensions of recent results (He 98) establishing Lyapunov stability of switched linear systems whose switching logics are generated by finite hybrid automata (Alur 94). In particular, we use unfoldings of the PTPN to identify equivalence classes of configurations from which fundamental cycles in the PTPN's reachability graph can be identified. The identified fundamental cycles are then used to form two different types of linear matrix inequalities whose feasibility ensure either the Lypaunov stability or the switched system's uniform ultimately bounded behaviour.

The remainder of this paper is organized as follows. Section 2 introduces the programmable timed Petri net. Section 3 uses the PTPN to model switched hybrid systems. Section 4 reviews prior results on switched system stability and uniform ultimate boundedness. In section 5, the proposed method for analyzing PTPN stability and performance is presented.

2. PROGRAMMABLE TIMED PETRI NETS

This section introduces an extension of the Alur-Dill hybrid system model (Alur 94) (Alur 96) in which timed Petri nets (Sifakis 77), rather than finite automata, generate the switching logic of the system. In particular, we introduce a *programmable timed Petri net* (PTPN) which is

a timed Petri net whose places, transtions, and arcs are all labeled with formulae representing constraints and reset conditions on the rates and times generated by a set of continuous-time systems called *clocks*.

An ordinary Petri net is a directed graph in which there are two types of nodes; places and transitions. Graphically, we represent the places by open circles and the transitions by bars. Petri nets are often characterized by the 4-tuple, (P,T,I,O) where P is the set of *places*, T is the set of *transitions*, $I \subset P \times T$ is a set of input arcs (from places to transitions), and $O \subset T \times P$ is a set of output arcs (from transitions to places). We denote the *preset* of a transition $t \in T$ as $\bullet t$ and define it as the set of places, $p \in P$ such that $(p,t) \in I$. In a dual manner, we introduce the postset of a transition $t \in T$ as $t\bullet$ and define it as the set of places, $p \in P$ such that $(t,p) \in O$.

The dynamics of ordinary Petri nets are characterized by the way in which the network mapping evolves. The marking $\mu : P \rightarrow Z$ is a mapping from the places onto non-negative integers. The marking $\mu(p)$ of place p denotes the number of *tokens* in that places (represented graphically by small filled circles). We say that the transition is *enabled* if $\mu(p) > 0$ for all $p \in \bullet t$. An enabled transition may *fire*. We introduce a firing function $q : T \rightarrow \{0,1\}$ such that $q(t) = 1$ if t is firing and is zero otherwise. If $\mu(p)$ and $\mu'(p)$ denote the marking of place p before and after the firing of enabled transition t, then

$$\mu'(p) = \begin{cases} \mu(p) + 1 & \text{if } p \in t\bullet / \bullet t \\ \mu(p) - 1 & \text{if } p \in \bullet t / t\bullet \\ \mu(p) & \text{otherwise} \end{cases} \qquad (1)$$

In ordinary Petri nets, places and transitions represent abstractions of the system "states" and "actions", respectively. In practice, however, we must remember that transitions (actions) take a finite amount of time to fire (complete). It is therefore necessary to work with timed Petri nets (Sifakis 77). In a timed Petri net the firing vector and marking vectors become functions of a global time τ. We denote the timed firing vector as q_τ. It indicates which transitions are in the act of "firing" at time τ. The timed marking vector is denoted as μ_τ. Just as in ordinary Petri nets, we will say that a transition t is enabled at time τ if $\mu_\tau(p) > 0$ for all $p \in \bullet t$. An enabled transition is free to fire. For the timed Petri net, however, the firing of a transition occurs over a time interval $[\tau_0, \tau_f]$. The length of this interval is called the transition's *holding time*. A transition t starts to fire at time τ_0 is said to be *committed* and its firing function $q_{\tau_0}(t)$ is set to unity. During the time that the transition is committed, the network's marking vector is not changed. It is only when the firing is completed at time τ_f that the marking vector is changed according to equation 1 given above. At the time the transition has completed firing, we also reset the firing function q_{τ_f} to zero.

The duration of the firing interval (holding time) can be characterized in a variety of ways. Common approaches assume that the holding time is either a fixed constant or a random variable. In some applications, there is a growing realization that these holding times can be treated as control variables. These times can be controlled by introducing "local" timers which fire when specified conditions *programmed* by the system designer are satisfied. This approach was used for concurrent state machines in (Alur 94). Essentially, this approach characterizes the holding times by logical propositions defined over the times generated by a set of *local clocks*. Petri nets whose holding times are defined in this way will be referred to as *programmable timed Petri Nets* (PTPN).

Let $N = (P,T,I,O)$ be an ordinary Petri net. We introduce a set, $\mathcal{X}$, of N *local clocks* where the ith clock $\mathcal{X}_i$ is denoted by the triple $(\dot{x}_i, x_{i0}, \tau_{i0})$. $x_{i0} \in \Re^n$ is a real vector representing the clock's offset. τ_{i0} is an initial time (measured with respect to the global clock) indicating when the local clock was started. $\dot{x}_i : \Re^n \rightarrow \Re^n$ is a Lipschitz continuous automorphism over $\Re^n$ characterizing the local clock's rate. Assume that the clock rate $\dot{x}_i$ is denoted by the automorphism f. The *local time* generated by the ith clock will be denoted as x_i which is a continuous function over $\Re^n$ that is the solution to the initial value problem,

$$\frac{dx_i}{dt} = f(x_i) \qquad (2)$$

$$x_i(\tau_{i0}) = x_{i0} \qquad (3)$$

for $\tau > \tau_{i0}$. We therefore see that the local timers are vector dynamical equations. The local time of the ith timer at global time τ is denoted as $x_i(\tau)$ and the timer's rate is denoted as $\dot{x}_i(\tau)$. We say that the *state* of the ith timer is the ordered pair $z_i(\tau) = (x_i(\tau), \dot{x}_i(\tau))$. The ensemble of all local clock states will simply be denoted as $z(\tau)$.

The interval $[\tau_0, \tau_f]$ over which a transition t will be firing is going to be characterized by formulae in a propositional logic whose atomic formulae are equations over the local times or clock rates of $\mathcal{X}$. An atomic formula, p, takes one of the following forms;

(1) It can be a *time constraint* of the form $f(x_i) = 0$ which means that the ith clock's state x_i is a zero for a known function $f : \Re^n \to \Re$.
(2) The atomic formula p can be a *rate constraint* of the form $\dot{x}_i = f$ which means that the ith clock's rate $\dot{x}_i$ is equal to the vector field $f : \Re^n \to \Re^n$.
(3) Finally, p can be a *reset equation* of the form $x_i(\tau) = \bar{x}_0$ which says that the ith clock's local time at global time τ is set to the vector $\bar{x}_0$.

We define a *well-formed formula* or wff as any expression where

- Any atomic formula is a wff,
- If p and q are wff's, then $p \wedge q$ is a wff.
- If p is a wff, then $\tilde{p}$ is a wff

The set of all wffs formed in this manner will be denoted as $\mathcal{P}$.

The *syntax* for well formed formulas is defined with respect to an underlying Petri net structure of the form $N = (P, T, I, O)$ and a set of local clocks $\mathcal{X}$. The local clock state z at time τ is said to *satisfy* a formula $p \in \mathcal{P}$ if p is "true" for the given clock state, $z(\tau)$. The satisfaction of p by $z(\tau)$ is denoted as $z(\tau) \models p$. The truth of the atomic formula is understood in the usual sense. We say that an atomic formla, $p \in \mathcal{P}$ is satisfied by $z(\tau)$ if and only if the evaluation of that formula is true. We say that $z(\tau) \models \tilde{p}$ if and only if $z(\tau)$ does not satisfy p. We say that $z(\tau) \models p \wedge q$ if and only if $z(\tau) \models p$ and $z(\tau) \models q$.

Consider an ordinary Petri net, $N = (P, T, I, O)$ and a set of logical timers, $\mathcal{X}$. A *programmable timed Petri net* (PTPN) is denoted by the ordered tuple, $(N, \ell_P, \ell_T, \ell_I, \ell_O)$ where; $\ell_P : P \to \mathcal{P}$, $\ell_T : T \to \mathcal{P}$, $\ell_I : I \to \mathcal{P}$, and $\ell_O : O \to \mathcal{P}$ label the places, transition, input arcs, and output arcs (respectively) of the Petri net N with a wff.

3. PTPN MODELING OF SWITCHED SYSTEMS

A hybrid dynamical system is a system which generates a mixture of event-driven (discrete-event) signals and continuous-valued signals. In this paper, we examine hybrid systems that can be viewed as a switched dynamical system, in which switches are generated when the plant's state crosses into specified regions of the state space. We can define such a hybrid system in a more formal manner as follows. Let $X \subset \Re^n$ and $Y \subset \Re^m$ be smooth n and m dimensional manifold and let I be a finite set of N integers. Let Δ be a constant dimensional distribution, $\Delta = \{f_1, \ldots, f_N\}$ where $f_i : X \times Y \to X$ for $i = 1, \ldots, N$ are locally Lipschitz vectorfields over X. We consider the system to be described by the following set of equations

$$\dot{x}(t) = f_{i(t)}(x(t), w(t)) \tag{4}$$
$$i(t) = q(x(t), i(t^-)) \tag{5}$$

where $x : \Re \to X$, $i : \Re \to I$, $f_i : X \to X$, $w : \Re \to Y$, and $q : X \times I \to I$. $i(t^-)$ refers to the left hand limit of the function $i(t)$ at point t. We refer to each f_i as a subsystem of the hybrid system. The signal, $w(t)$ represents a disturbance whose essential supremum satisfies a specified bound. In particular, we say that $w \in \mathcal{BL}_\infty$ if $\operatorname{ess\,sup} \|w(t)\| < 1$. The preceding model is essentially that used in (Tavernini 87) with the addition of the bounded exogenous disturbance, $w(t)$.

The *trajectory* of the switched system is the ordered pair, (x, i), where $x : \Re \to X$ and $i : \Re \to I$ which *solves* the system equation assuming that the disturbance $w(t)$ is known. The value taken by the trajectory at time $t \in \Re$ is denoted by $(x(t), i(t))$. We say that (x, i) *solves* the system equation if and only if the equations are satisfied by $x(t)$ and $i(t)$ for all $t \in \Re$ and some $w \in \mathcal{BL}_\infty$. This paper does not treat questions concerned with the existence of solutions. In general, however, solutions (when they do exist) will not be unique due to the nondeterminism of the switching law.

The preceding model of a switched system assumes a very general switching function, q. To obtain more precise results, however, we need to specify the nature of the switching function. In this section we examine the use of the PTPN in modeling the switched system. This is accomplished by viewing each subsystem of the switched system as a local timer in the PTPN. The switching rules are then embedded in the PTPN through the labeling functions ℓ_P,ℓ_T, ℓ_I, and ℓ_O.

We begin by viewing the state of the dynamical systems x_i as local times and the vector fields f_i are viewed as clock rates for these timers. The switching logic is represented by the ordinary Petri net structure $N = (P, T, I, O)$ and the labeling functions are chosen as follows,

- $\ell_P(p)$ is chosen to be an atomic rate formula of the form $\dot{x}_i = f_j$ where f_j is one of the vector fields in Δ. When this place is marked

then the system changes the ith clock rate to f_j.

- $\ell_T(t)$ is chosen to be a tautology. If there are constraints on the various subsystems during the firing of a transition, this is where those constraints would be placed. For instance, we could have another timer here, which is reset when the transition is first committed to firing.
- $\ell_I((p,t))$ is chosen to be a wff whose truth commits the transition t to firing.
- $\ell_O((t,p))$ is chosen as a wff whose truth completes the firing of transition t.

Let (x,i) be the trajectory generated by a switched dynamical system. We let the set of switching times of (x,i) be those times when $\lim_{\tau\to t^+} i(\tau) \neq \lim_{\tau\to t^-} i(\tau)$. The sequence of switching events of (x,i) is the sequence of $i(t_k)$ (for $k = 1,\cdots,\infty$) where t is a switching time. The trajectory is said to be *deadlock free* if the sequence of switching events is not finite. We say that the trajectory is *live* if the event sequence contains an infinite number of each index, $i \in I$. We say that the trajectory is *nonZeno* if the sequence of switching times is not summable (i.e. $\sum_n t_n > \infty$ where t_n are switching times). Note that an important issue which is not addressed in this paper concerns necessary and sufficient conditions for a switched system to be live, deadlock free, or nonZeno. In this paper, we assume that the switched system is live and nonZeno.

We now present a specific example illustrating the use of a PTPN in modeling the switching behaviour of a power system. Consider the power system shown in figure (1). There are four nodes(Generators) in this system. Nodes 1,2,4 are generator nodes and node 3 is reference node. Let $\theta_i, i = 1,2,4$ denote the generator rotor angle of node 1,2,4. Let $\theta_i, \dot{\theta}_i, i = 1,2,4$ represent the state space of the power system. It can be shown that the state space of the system evolves according to the following set of differential equations:

$$\dot{x} = f(x(t), w(t)) \tag{6}$$

where $x = [\theta_1, \dot{\theta}_1, \theta_2, \dot{\theta}_2, \theta_4, \dot{\theta}_4]^T$ is the state space, $\omega \in \Re^3$ is the disturbance satisfying $\|\omega\| \leq 1$, $f : \Re^6 \times \Re^3 \to \Re^6$ is a continuous and locally Lipschitz mapping.

Linearize equation 6 at the equlibrium point x_0, to obtain

$$\dot{x} = Ax(t) + Bw(t) \tag{7}$$

where

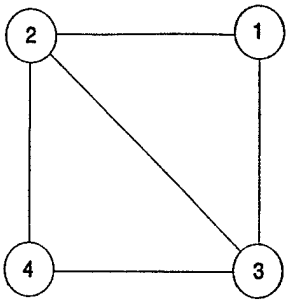


Fig 1. The example power system

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ -B_{11} & -D_1 & -B_{12} & 0 & -B_{14} & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ -B_{21} & 0 & -B_{22} & -D_2 & -B_{24} & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ -B_{41} & 0 & -B_{42} & 0 & -B_{44} & -D_4 \end{bmatrix} \tag{8}$$

$$B = \begin{bmatrix} 0\,0\,0 \\ 1\,0\,0 \\ 0\,0\,0 \\ 0\,1\,0 \\ 0\,0\,0 \\ 0\,0\,1 \end{bmatrix} \tag{9}$$

where $B_{ij}, i,j = 1,2,4$, represent the parameter of the transmission lines, $D_i, i = 1,2,4$ represent the winding ratio of generator attached to node i. Let $z = [\theta_1, \theta_2, \theta_3]$

The control objective is to let $\|z(t)\|_\infty \leq 0.1$, for all $t \in [0,\infty]$. To help achieve this goal we introduce the following supervision policy. Assumming each generator has two winding ratio to choose from. D_{i0} and $D_{i1}, i = 1,2,4$, where $D_{i0} = 2D_{i1}$. We say node i is in mode 0 if $D_i = D_{i0}$, and in mode 1 if $D_i = D_{i1}$. Each generator node obeys the following local switching rule.

(1) If a fault is detected in the local neighbourhood of node i and node i is currently in mode 1, then switch it to mode 0. This will protect the generator from suffering large transient oscillation. The fault is detected when $|\theta_i| > 0.05$.

(2) if a request of changing the load condition of node i is generated and node i is currently in mode 0, then switch it to mode 1. This will ensure the fast-adaptation of the generator to new load condition. In this system, we assume that the request for load change is issued 10 seconds after the fault was tripped. We model this by labeling the output arcs with a bound on a resetable timer, τ_i.

The timer, $(1, \tau_i, \tau_{i0})$, which is used to reset system i after a fault, is assumed to have a rate of unity. It is reset when the fault is tripped.

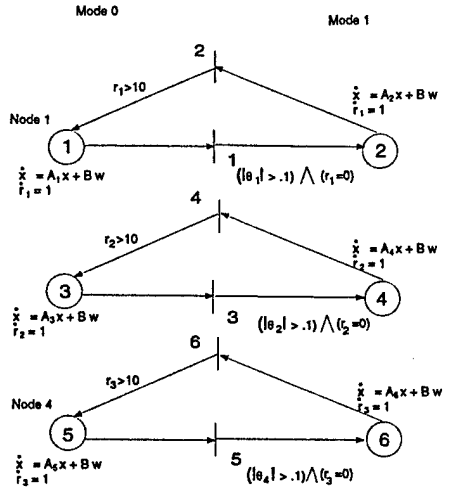


Fig 2. Petri-net model of power system

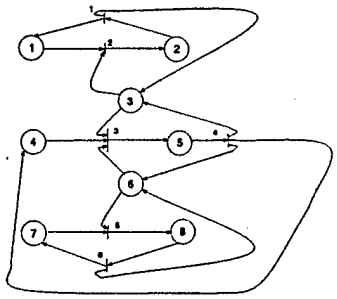


Fig 3. The controlled Petri-net for power system

We model the reset by labeling the output arc of the PTPN with the equation $r_i = 0$. The PTPN of the power system based on the above local switching rule is shown in figure (2).

Simulation of the operation of the power system shows if two neighbouring generator nodes, i.e. nodes 1,2 or node 2,4 are both in mode 1, the rotor angle of the generators will have large variation (> 0.1)under disturbance ω, which violates our control objective. To achieve the control objective, we thus implement supervisory control logic to prohibit the previously-mentioned cases from happening. The Petri-net model of the controlled system is shown in figure (3). This supervision introduces a place between two "adjacent" generators which enforces a mutual exclusion condition. It is this supervised system whose stability will be studied in following sections.

4. HYBRID AUTOMATON STABILITY AND PERFORMANCE

Lyapunov stability and uniform ultimate boundedness are standard concepts in the study of nonlinear dynamical systems. Consider a disturbed system which can be represented by the differential equation, $\dot{x} = f(x, w)$ where $w \in BL_\infty$.We say that x_0 is an *equilibrium point* of the undisturbed system if $f(x_0, 0) = 0$. The equilibrium point of the undisturbed system is said to be stable in the sense of Lyapunov if for all $\epsilon > 0$ there exists $\delta > 0$ such that $x(t_0) < \delta$ implies that $x(t) < \epsilon$ for all $t > t_0$. We say that the disturbed system is uniformly ultimately bounded if and only if for all $\epsilon > 0$ there exists a time $T(\epsilon) > 0$ such that if $x(t_0) < \epsilon$, then $x(t) < \delta$ for all $t > T(\epsilon)$.

There are a variety of results providing sufficient conditions for the Lyapunov stability of switched (undisturbed) systems. In (Peleties 91) a single positive definite functional is found which is a Lyapunov function for all subsystems of the switched system. Multiple Lyapunov functionals methods (Branicky 94) (Hou 96) have been developed which apply to a larger set of systems than the single Lyapunov function methods. In certain cases, where the switched system consists of linear time invariant subsystems, it has been suggested that multiple candidate Lyapunov functions can be determined by finding feasible points of a linear matrix inequality (LMI) (Petterson 96). These last results are particularly important because they provide a computational method for checking the sufficient conditions for switched system stability provided in (Branicky 94).

The sufficient conditions presented in (Branicky 94) (Hou 96) and used in (Petterson 96) to compute candidate Lyapunov functionals provide a very conservative approach for testing switched system stability. In the first place, the stability theorems in (Branicky 94) (Hou 96) require that V_j be Lyapunov like for all possible switching sequences. These papers place no assumptions on the nature of the switching laws used so that when the computational methods of (Petterson 96) are employed, the worst case switching law has to be considered. The worst case switching law is one in which every switch is possible. This assumption can result in an extremely high dimensional linear matrix inequality which may be more restrictive than it needs to be.

In (He 98), it was shown that the LMI's formulated in (Petterson 96) could be simplified significantly when the system's switching logic is generated by a finite automaton. Rather than requiring Lyapunov-like functions over every possible switching sequence, it suffices to consider *fundamental cycles* of the automaton. Recall that a cycle is any path accepted by the automaton which starts and ends at the same vertex. If the cycle does not cross itself, then the cycle

is *fundamental.* Before stating this theorem, we need to introduce some preliminary definitions.

Let the subsequence of times when system j is turned on and off be denoted as

$$\lambda_j = t_{n_1}, t_{n_1+1}, \cdots, t_{n_k}, t_{n_k+1}, \cdots \quad (10)$$

Define the interval completion $I(\lambda_j)$ as the set obtained by taking the union of all open intervals when system j is active. Denote $E(\lambda_j)$ as the subsequence of times when system j is turned on. We say that a continuously differentiable function $V : \Re^n \rightarrow \Re^+$ is *Lyapunov-like function* over set of switching times, λ, if and only if $\dot{V}(x(t)) \leq 0$ for all $t \in \mathcal{I}(\lambda)$ and V is monotonically nonincreasing on $E(\lambda)$. We can now state the following theorem from (He 98).

Theorem 1. Consider a live nonZeno switched system where $f_j(0) = 0$ for all $j \in I$. Let λ be a switching sequence generated by the system. Let μ denote a subsequence of contiguous switches in λ such that the event sequence is a fundamental cycle of the system's automaton. Let $\bar{\mu}$ denote the infinite sequence formed by concatenation of μ with itself. If there exist a set of continuously differentiable functions $V_j : \Re^n \rightarrow \Re$ which are Lyapunov like over sequence $\bar{\mu}_j$ for all $j \in I$, then the system is stable in the sense of Lyapunov.

By combining the preceding theorem with the LMI methods in (Petterson 96), we obtain a computationally efficient method for determining if a switched LTI system is Lyapunov stable. A similar set of conditions can also be used to establish sufficient conditions for a switched LTI system with bounded exogenous disturbances to possess uniform ultimate bounded performance. The key result here is the switching lemma stated in (Bett 97). In this theorem we say that a matrix $P \in \text{FeasRic}(A, B, \alpha, \beta)$ if P satisfies the following Riccati inequality,

$$A'P + PA + (\alpha + \beta)P + \frac{1}{\alpha}PBB'P \leq 0 \quad (11)$$

With this definition, we can now state the theorem proven in (Bett 97)

Theorem 2. Consider two LTI systems $\Sigma_1 = (A_1, B_1, C_1, D_1)$ and $\Sigma_2 = (A_2, B_2, C_2, D_2)$ and consider any finite constants $r \in (0, 1]$ and $\gamma > 0$. Suppose there exists positive constants α, β, and ρ and positive definite matrices P_1 and P_2 such that

$$rP_2 \leq P_1 \quad (12)$$

$$\gamma^2 P_1 \geq C_1' C_1 \quad (13)$$

$$\gamma^2 P_2 \geq C_2' C_2 \quad (14)$$

$$P_1 \in \text{FeasRic}(A_1, B_1, 2\beta + \frac{\alpha}{r}, \alpha) \quad (15)$$

$$P_2 \in \text{FeasRic}(A_2, B_2, \rho, \rho) \quad (16)$$

Consider a time $t_s > 0$ and let w, x, and z be the input, state, and output of the dynamical system which evolves according to system Σ_1 for $t_0 < t < t_s$ and which evolves according to Σ_2 for $t > t_s$. If

$$t_s > -\frac{1}{2\beta}\log r = t_d \quad (17)$$

then $\|z\|_\infty \leq \gamma$ for all $t > t_d$. We call t_d the switched system's delay time.

The preceding theorem establishes conditions that the LTI system needs to satisfy to ensure uniform ultimate bounded behaviour. Note that these conditions are also linear matrix inequalities, similar in structure to those used by (Petterson 96) The obvious implication here is that we should able be to easily state a result similar to the fundamental cycle result in (He 98) to test for uniform ultimate boundedness.

5. PTPN STABILITY AND PERFORMANCE

This section uses the results of section 4 to propose a method for analyzing the stability and performance of switched LTI systems represented by PTPN. The principal observation is that if the PTPN is live nonZeno and bounded, then the reachability graph of the PTPN is finite. In particular, this reachability graph now forms an automaton model for the system's switching logic. We can then use the results of the preceding section to provide sufficient tests for hybrid system stability and uniform ultimate bounded performance. The problem here however, is that the construction of the reachability graph for the PTPN is not a simple computation. The computational complexity associated with constructing the reachability graph grows exponentially with the number of places and transitions in the Petri net. A more appropriate way to identify the fundamental cycles of the PTPN is to use a partial order method such as an unfolding (Engelfreit 91). Constructing a complete finite prefix of a PTPN unfolding (McMillan 92) has been shown to require a polynomial number of computations.

Given a Petri net, we say that two places p_1 and p_2 are in conflict if there exist distinct transitions

t_1 and t_2 such that $\bullet t_1 \cap \bullet t_2 \neq \emptyset$ and (t_1, p_1) and (t_2, p_2) belong to the reflexive and transitive closure of the arcs. In other words, p_1 and p_2 are in conflict if two paths terminating at those places start at the same place and immediately diverge.

An occurence net is an acyclic conflict free net. A branching process of network N is a pair (N', h) such that N' is an occurrence net and h is a net homomorphism mapping N' to N in a way that preserves the behaviour of the original net (see (Englefriet 91) or (Esparza 96) for precise definitions). An unfolding is the maximal branching process associated with a Petri net N. Consider the occurrence net of a Petri net unfolding. A configuration C of this net is a set of transitions satisfying the following conditions,

- $t \in C$ implies for all $t' \leq t : t' \in C$
- for all transitions in C are conflict free.

An occurence net may have several different configurations. Two configurations which can be marked at the same time are said to be "concurrent". Concurrency can be viewed as an equivalence relation over the set of all configurations of an occurrence net. In particular, this mean that the set of configurations can be partitioned into equivalence classes.

The unfolding of a PTPN can be used to study the stability and performance of a switched system represented by a PTPN. The result cited above from (He 98) asserts that a sufficient condition for the stability (Lyapunov) and ultimate bounded behaviour of a switched LTI system is that a set of LMI's associated with the fundamental cycles of the system's reachability graph be feasible. These fundamental cycles can be found by constructing the reachability tree; an inefficient approach, or they can be systematically constructed from the network unfolding. In the following example, we illustrate how the use of unfoldings can dramatically reduce the complexity of searching for fundamental cycles in the PTPN's reachability graph.

Now let's go back to the example. The unfolding of the controlled Petri-net is shown in figure (4). We identify 3 configurations in the unfolding result. Here we describe them by three sets of transitions, namely, $(1,2), (3,4), (5,6)$. We label the three configurations as c_1, c_2, c_3, respectively. Concurrency of these configurations induces two equivalence classes. Configuration c_2 is associated with one of these equivalence classes and configurations c_1 and c_3 form the other equivalence class. We have been able to

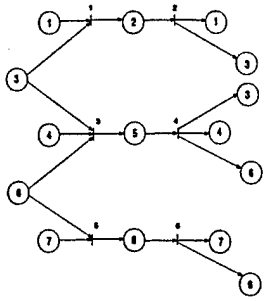


Fig 4. Occurrence Network of Problem

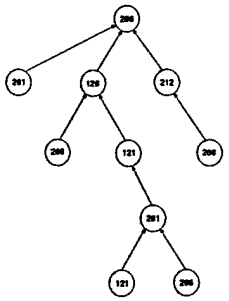


Fig 5. Reachability Graph

develop a systematic algorithm for constructing the fundamental cycles. In this particular example, the second configurational equivalence class has 4 fundamental cycles represented by the sequence of firing of transitions 1, 2, 5, 6. These fundamental cycles are 1-2, 5-6, 1-5-6, 5-1-2. For each fundamental cycle, a set of LMI is established. Solving the LMI problem shows the system satisfies the bounded amplitude objective, i.e. $\|z\|_{\infty,[0,\infty]} < 0.1$. Simulation results have validated the correctness of this approach.

In comparison, the reachability tree of this example is also established. The results shows, to build the reachability tree, a total number of 10 nodes has to be created, 33 paths have to be traced and 85 calculations have to be made in tracing all the paths. While in the unfolding process, only 3 configurations are found, 3 paths are traced and 20 calculation are made in tracing all these paths. This fact clearly demonstrates that unfolding provides a more efficient method for finding fundamental cycles in PTPN. This empirical finding supports the claims made in (McMillan 92) where it was asserted that the computational complexity of constructing the reachability graph is exponential in the number of places and transitions. In contrast, the unfolding generally has a polynomial complexity. This difference is illustrated quite graphically by simply comparing the reachability graph for this problem (see figure (5)) and comparing it to the relatively simple occurrence net for this problem (figure (4)).

6. SUMMARY

This paper has reported on some preliminary findings concerned with the use of Petri nets in the analysis of switched hybrid systems. In particular, we introduced an extension of the Alur-Dill hybrid automaton which we refer to as the programmable timed Petri net. We then showed how the PTPN can be used to model switched hybrid systems. Using earlier results from (He 98) and (Bett 97), it was asserted that the stability and uniform ultimate boundedness for the switched linear systems could be guaranteed provided a set of Linear Matrix Inequalities (LMI) was shown to be feasible. This result is similar to that found in (Petterson 96). The difference between our results and these earlier results, is that (He 98) shows that it is sufficient to form the LMI's over the fundamental cycles of the system's reachability graph. This leads to an enormous simplification of the resulting analysis effort. This paper applied the results of (He 98) to the PTPN by using unfoldings of the PTPN to systematically and efficiently search for the fundamental cycles of the Petri net.

There are a variety of directions for future research. In the first place, the results presented here are sufficient conditions. For system verification it is important to have necessary and sufficient conditions. While pure necessity may be difficult to obtain, it should be possible to obtain very tight sufficient conditions through optimization methods. In addition to this, it is important to study necessary and sufficient conditions for the PTPN to be live and nonZeno. Such conditions require an analysis of the system's viability (Deshpande 95). We suspect that unfoldings may once again prove to be important in managing the complexity associated with finding such viability kernels.

7. REFERENCES

[1] R. Alur and D.L. Dill, A Theory of Timed Automata, *Theoretical Computer Science*, 126:183-235, 1994

[2] R. Alur, T. Henzinger, and P-H Ho., Automatic Symbolic Verification of Embedded Systems, *IEEE Transactions on Software Engineering*, 22:181-201, 1996.

[3] C.J. Bett and M.D. Lemmon, Bounded Amplitude Control using Multiple Linear Agents, Technical Report ISIS-97-004, Dept. of Electrical Eng, University of Notre Dame, March 1997

[4] M. Branicky, Stability of Switched and Hybrid Systems, In *33rd Conference on Decision and Control*, 1994.

[5] A. Deshpande and P. Varaiya, Viable Control of Hybrid Systems, *Hybrid Systems II*, A. Nerode (ed.), LNCS Volume 999, Springer-Verlag, 1995.

[6] J. Engelfriet, "Branching Processes of Petri Nets", *Acta Informatica*, 28, pp 575-591, 1991.

[7] J Esparza, St. Romer, and W. Vogler, "An improvement of McMillan's unfolding algorithm", TACAS'96, Passau, March 1996, LNCS Vol 1055, Springer-Verlag, 1996.

[8] Kevin X He, M.D. Lemmon, "Lyapunov Stability of Continuous Valued Systems Under the Supervision of Discrete Event Transition Systems", Technical Report of the ISIS Group, University of Notre Dame, ISIS-97-010, October 1997,

[9] L. Hou, A.N. Michel, and H. Ye, Stability Analysis of Switched Systems, In *35th Conference on Decision and Control*, December 1996.

[10] K. McMillan. Using unfoldings to avoid the state explosion problem in the verification of asynchronous circuits, CAV'92, LNCS 663 Springer-Verlag, 1992.

[11] P. Peleties and R. DeCarlo, Asymptotic Stability of m-switched systems using Lyapunov-like functins. *American Control Conference*, June 1991.

[12] S. Pettersson and B. Lennartson, Stability and Robustness for Hybrid Systems, In *35th Conference on Decision and Control*, 1996.

[13] J. Sifakis. Use of petri nets for performance evaluation. In *Measuring, modelling and evaluating computer systems.* North Holand, 1977.

[14] L. Tavernini, Differential automata and their discrete simulators, *Nonlinear Analysis, theory, methods, and applications*, 11(6):665-683, 1987.

APPENDIX H

J.O. Moody and P.J. Antsaklis
"Supervisory Control Using Computationally Efficient Linear Techniques: A Tutorial Introduction"
Proc of 5th IEEE Mediterranean Conference on Control and Systems
Paphos, Cyprus, July 21-23, 1997. Also released as ISIS Technical Report isis-97-015, October 1997.

Supervisory Control Using Computationally Efficient Linear Techniques: A Tutorial Introduction

Technical Report of the ISIS Group
at the University of Notre Dame
ISIS-97-015
October, 1997

John O. Moody and Panos J. Antsaklis
Department of Electrical Engineering
University of Notre Dame
Notre Dame, IN 46556

Interdisciplinary Studies of Intelligent Systems

Supervisory Control Using Computationally Efficient Linear Techniques: A Tutorial Introduction[1][2]

Abstract

This paper provides an overview of a computationally efficient method for synthesizing supervisory controllers for discrete event systems (DES). The DES plant and controller are described by Petri nets which provide a useful linear algebraic model for both control analysis and synthesis. It is shown how a set of linear constraints on the plant's behavior can be enforced, accounting for possibly uncontrollable or unobservable transitions in the plant net, using techniques from Petri net theory, integer programming, and linear systems. The paper is written as a tutorial introduction to the approach. Several results presented here have been reported elsewhere in the literature.

1 Introduction

A methodology to automatically derive feedback supervisory controllers for discrete event systems (DES) described by Petri nets appears in [13]. The control designer is presented with a Petri net model of a DES and a set of linear constraints on the state space of the DES. The control goal is to insure that the constraints are met during the plant's normal operation. In the spirit of supervisory control, this task is accomplished by prohibiting certain occurrences in the plant which would cause one or more of the constraints to be violated. The method is based on the idea that specifications representing desired plant behaviors can be enforced by making them invariants of the controlled Petri net. The resulting controllers are themselves Petri nets and are identical to the monitors [2] of Giua et al. The controller's size is proportional to the number of constraints.

The supervisor is used to enforce a set of linear constraints on the state space of the plant DES. These constraints are not as general as the languages enforced by Ramadge and Wonham [10] in their work on supervisory control using automata, but the solution algorithms are simpler, and they can be used to describe a broad variety of problems including

- A large range of forbidden state problems.
- Serial, parallel and general mutual exclusion problems.
- A class of logical predicates on plant behavior [12].

[1]This technical report has also appeared as J. O. Moody and P. J. Antsaklis, "Supervisory control using computationally efficient linear techniques: A tutorial introduction", In *Proceedings of the 5th IEEE Mediterranean Conference on Control and Systems*, Session MP1, Paphos, Cyprus, July 1997.

[2]This research was partially funded by the National Science Foundation. Grant ECS95–31485.

- Conditions involving the occurrence of events and particular regions of the state space.
- Conditions involving the concurrence of events.
- The modeling of shared resources [6].

The approach was extended in [7] to apply to Petri nets which contain uncontrollable transitions, the firing of which cannot be inhibited by the controller. This work was partially motivated by the research of Li and Wonham [3] dealing with the enforcement of linear constraints on vector discrete event systems with uncontrollable events. The approach in [7] was expanded in [4] to include uncontrollable and unobservable transitions in a unified framework. Algorithms were presented for automatically computing new sets of plant constraints which accounted for uncontrollable and unobservable transitions while still enforcing the original constraints. Unobservable transitions force a special structure on the Petri net controller which can be used to characterize valid controllers and simplify controller design. These results appear in [5]. These contributions extend the applicability of the control method while maintaining its original emphasis: they also relate to Petri net place invariants and are again simple to implement with excellent numerical properties.

The paper is structured as follows. The controller synthesis method for plants with controllable transitions is described in section 2. A methods for dealing with uncontrollable and unobservable transitions is covered in section 3. An example is used to illustrate the method in section 4, and concluding remarks are given in section 5.

2 Automatic Controller Synthesis

The system to be controlled is modeled by a Petri net with n places and m transitions and is known as the process or *plant net.* The incidence matrix of the plant net is D_p. It is assumed that all the enabled transitions can fire. It is possible that the process net will violate certain constraints placed on its behavior, thus the need for control. The *controller net* is a Petri net with incidence matrix D_c made up of the process net's transitions and a separate set of places. The *controlled system* or *controlled net* is the Petri net with incidence matrix D made up of both the original process net and the added controller. The control goal is to force the process to obey constraints of the form

$$L\mu_p \leq b \tag{1}$$

where μ_p is the marking vector of the Petri net modeling the process, L is an $n_c \times n$ integer matrix, b is an n_c dimensional integer vector and n_c is the number of constraints. Note that the inequality is with respect to the individual elements of the two vectors $L\mu_p$ and b and can be thought of as the logical conjunction of the individual "less than or equal to" constraints. This definition will be used throughout this paper whenever vectors appear on either side of an inequality sign.

Inequality (1) can be transformed into an equality by introducing an external Petri net controller which contains places which represent nonnegative "slack variables" μ_c. The constraint then becomes

$$L\mu_p + \mu_c = b \tag{2}$$

where μ_c is an n_c dimensional integer vector which represents the marking of the controller places. Note that $\mu_c \geq 0$ because the number of tokens in a place can not become negative; thus equation (2) implies inequality (1). The controller places insure that the weighted sums of tokens in the process net's places are always less than or equal to the elements of b. The places which maintain the inequality constraints are part of a separate net called the controller net. The structure of the controller net will be computed by observing that the introduction of the slack variables forces a set of place invariants on the overall controlled system defined by equation (2).

Place invariants are one of the structural properties of Petri nets. See [8, 9, 11] for more information on Petri nets and their properties and analysis. A place invariant is defined as every integer vector x which satisfies

$$x^T\mu = x^T\mu_0 \text{ (a constant)} \tag{3}$$

where μ_0 is the net's initial marking, and μ represents any subsequent marking. Equation (3) means that the weighted sum of the tokens in the places of the invariant remains constant at all markings and this sum is determined by the initial marking of the Petri net. The place invariants of a net are elements of the kernel of the net's incidence matrix, i.e., they can be computed by finding integer solutions to

$$x^T D = 0 \tag{4}$$

where D is an $n \times m$ incidence matrix with n being the number of places and m the number of transitions.

The matrix D_c contains the arcs that connect the controller places to the transitions of the process net. Let $\mathbb{Z}$ be the set of integers. The incidence matrix $D \in \mathbb{Z}^{(n+n_c)\times m}$ of the closed loop system is given by

$$D = \begin{bmatrix} D_p \\ D_c \end{bmatrix} \tag{5}$$

and the marking vector $\mu \in \mathbb{Z}^{n+n_c}$ and initial marking μ_0 are given by

$$\mu = \begin{bmatrix} \mu_p \\ \mu_c \end{bmatrix} \qquad \mu_0 = \begin{bmatrix} \mu_{p_0} \\ \mu_{c_0} \end{bmatrix} \tag{6}$$

Note that equation (2) is in the form of (3), thus the invariants defined by equation (2) on the system (5), (6) must satisfy equation (4).

$$\begin{aligned} X^T D = [L\ I] \begin{bmatrix} D_p \\ D_c \end{bmatrix} &= 0 \\ LD_p + D_c &= 0 \end{aligned} \tag{7}$$

where I is an $n_c \times n_c$ identity matrix since the coefficients of the slack variables in equation (2) are all equal to 1. The following proposition follows from this discussion.

Proposition 1. The Petri net controller, $D_c \in \mathbb{Z}^{n_c \times m}$ with initial marking μ_{c_0}, which enforces constraints (1) when included in the closed loop system (5) with marking (6) is defined by

$$D_c = -LD_p \tag{8}$$

with initial marking

$$\mu_{c_0} = b - L\mu_{p_0} \tag{9}$$

assuming that the transitions with arcs from D_c are controllable, observable, and that $\mu_{c0} \geq 0$.

The controller defined by proposition 1 is maximally permissive, assuming that all transitions are controllable and observable, in that it will never disable a transition that would not directly violate the constraints if fired. The proof of this result is given in [13].

Proposition 1 creates a controller which will enable and inhibit various transitions in the plant. If any of these transitions are uncontrollable or unobservable, then the controller defined by this method may be invalid. The next section shows how a transformation of the constraints can be performed in order to avoid these transitions while still enforcing the original constraints.

3 Handling Uncontrollable and Unobservable Transitions

Consider the situation where the controller is not allowed to influence certain transitions in the plant Petri net. These transitions are called uncontrollable. It is illegal for the Petri net controller to include an arc from one of the controller places to any of these uncontrollable plant transitions, since these kinds of connections can lead to the disabling of plant transitions.

Equation (8) in section 2 shows that it is possible to construct the incidence matrix D_c of a maximally permissive Petri net controller as a linear combination of the rows of the incidence matrix of the plant. Negative elements in D_c correspond to arcs from controller places to plant transitions. These arcs act to inhibit plant transitions when the corresponding controller places are empty, and thus they can only be applied to plant transitions which permit such external control. Group all of the columns of D_p which correspond to transitions which can not be controlled into the matrix D_{uc}. The matrix LD_{uc} must contain no positive elements[3], as these will correspond to controlling arcs when constructing the supervisor

[3]Actually LD_{uc} may contain positive elements when the controller is merely observing uncontrollable transitions and not inhibiting them, but this situation is not covered here.

$D_c = -LD_p$. An enforceable set of constraints will satisfy

$$LD_{\text{uc}} \leq 0 \tag{10}$$

It is also possible that transitions within the plant may be unobservable, i.e., they are defined on the Petri net graph because they represent the occurrence of real events, but these events are either impossible or too expensive to detect directly. It is also possible, in the event of a sensor failure, that a transition might suddenly become unobservable, forcing a redesign or adaptation of the control law. It is illegal for the controller to change its state based on the firing of an unobservable transition, because there is no direct way for the controller to be told that such a transition has fired. Both input and output arcs from the controller places are used to change the controller state based on the firings of plant transitions. Let the matrix D_{uo} represent the incidence matrix of the unobservable portion of the Petri net. This matrix is composed of the columns of D_p which correspond to unobservable transitions, just as D_{uc} is composed of the uncontrollable columns of D_p. It is illegal for the controller $D_c = -LD_p$ to contain any arcs in the unobservable portion of the net, thus an enforceable set of constraints will satisfy

$$LD_{\text{uo}} = 0 \tag{11}$$

Conditions (10) and (11) indicate that it is possible to observe a transition that we can not inhibit, but it is illegal to directly inhibit a transition that we can not observe.

Suppose, given a set of constraints $L\mu_p \leq b$, we construct the matrices LD_{uc} and LD_{uo} and observe that there are violations to conditions (10) and/or (11). Since the controller is made of a linear combination of the rows of D_p, it is interesting to consider the situation where we use the addition of further rows from D_{uc} in order to eliminate the positive elements of LD_{uc} and use rows from D_{uo} to eliminate the nonzero elements of LD_{uo}, i.e., if we are going to use a place invariant forming Petri net controller, what additions to the constraints would we need to make in order to eliminate positive elements from LD_{uc} and nonzero elements from LD_{uo}? What constraints, of the form $L'\mu_p \leq b'$, that can be enforced by an invariant-based controller, will also maintain the original constraint $L\mu_p \leq b$ while not interfering with the uncontrollable/unobservable portions of the plant? The following lemma appeared in [7].

Lemma 2.

$$\text{Let } R_1 \in \mathbb{Z}^{n_c \times m} \text{ satisfy } R_1\mu_p \geq 0 \ \forall \ \mu_p. \tag{12}$$

$$\text{Let } R_2 \in \mathbb{Z}^{n_c \times n_c} \text{ positive definite diagonal matrix} \tag{13}$$

If $L'\mu_p \leq b'$ where

$$L' = R_1 + R_2 L \tag{14}$$

$$b' = R_2(b+1) - 1 \tag{15}$$

and $\mathbf{1}$ is an n_c dimensional vector of 1's, then $L\mu_p \leq b$.

Lemma 2 shows a class of constraints, $L'\mu_p \leq b'$, which, if enforced, will imply that $L\mu_p \leq b$ are also enforced. In [4], Lemma 2 is used to prove a portion of the following proposition.

Proposition 3. Let a plant Petri net with incidence matrix D_p be given with a set of uncontrollable transitions described by D_{uc} and a set of unobservable transitions described by D_{uo}. A set of linear constraints on the net marking, $L\mu_p \leq b$, are to be imposed. Assume R_1 and R_2 meet (12) and (13) with $R_1 + R_2 L \neq 0$ and let

$$\begin{bmatrix} R_1 & R_2 \end{bmatrix} \begin{bmatrix} D_{\text{uc}} & D_{\text{uo}} & -D_{\text{uo}} & \mu_{p_0} \\ LD_{\text{uc}} & LD_{\text{uo}} & -LD_{\text{uo}} & L\mu_{p_0} - b - \mathbf{1} \end{bmatrix} \leq \begin{bmatrix} 0 & 0 & 0 & -\mathbf{1} \end{bmatrix} \tag{16}$$

Then the controller

$$D_c = -(R_1 + R_2 L)D_p = -L'D_p \tag{17}$$

$$\mu_{c_0} = R_2(b+1) - \mathbf{1} - (R_1 + R_2 L)\mu_{p_0} = b' - L'\mu_{p_0} \tag{18}$$

exists and causes all subsequent markings of the closed loop system (5),(6) to satisfy the constraint $L\mu_p \leq b$ without attempting to inhibit uncontrollable transitions and without detecting unobservable transitions.

The usefulness of proposition 3 for specifying controllers to handle plants with uncontrollable and unobservable transitions lies in the ease in which the matrices R_1 and R_2, with the appropriate properties, can be generated. Algorithms for solving this problem including a method involving matrix row operations and by solving a linear integer programming problem appear in [4]. The method of using row operations is outlined below, but instead of presenting the pseudo code algorithms of [4], the overall motivation and goals of the method are described.

To meet assumption (12), it is sufficient to assume that all of the elements of R_1 are nonnegative, since the elements of μ_p are nonnegative by definition. In general, for unbounded μ_p, it is necessary that all of the elements of R_1 be nonnegative, however if bounds on μ_p are known, then it is possible to generate valid R_1 vectors which contain some negative elements. If R_1 and R_2 which satisfy (12) and (13) do exist, then they can be found by performing row operations on $\begin{bmatrix} D_{\text{uc}} \\ LD_{\text{uc}} \end{bmatrix}$ and $\begin{bmatrix} D_{\text{uo}} \\ LD_{\text{uo}} \end{bmatrix}$. Row operations act as premultiplications of a matrix, just as $\begin{bmatrix} R_1 & R_2 \end{bmatrix}$ premultiplies these two matrices in inequality (16). R_1 and R_2 can be found by finding a set of row operations which do not involve premultiplication of any row by a negative number and which force the LD_{uc} portion of the matrix to contain all zero or negative elements and the LD_{uo} matrix to be all zeros. Note that assumption (13), which requires R_2 to be a positive definite matrix, is not restrictive. This matrix simply represents the premultiplication coefficients of the rows of the LD_{uc} and LD_{uo} portions of

the matrices undergoing row operations. We can assume this matrix is diagonal because LD_{uc} and LD_{uo} are linearly dependent with D_{uc} and D_{uo}, i.e., we will never need to take linear combinations of the rows in LD_{uc} or LD_{uo}. We can also assume that the diagonal elements are positive since, if negative numbers are required, they can be accounted for by R_1, which still needs to meet assumption (12). This technique is illustrated for a plant with uncontrollable transitions in the following section.

4 Example – The Unreliable Machine

We now provide a simple example in order to illustrate the concepts that have been covered above. The example plant is partially based on the model of an "unreliable machine" from [1]. The machine is used to process parts from an input queue, completed parts are moved to an output queue. The machine is considered unreliable because it is possible that it may break down and damage a part during operation. This behavior is captured in the plant model. Damaged parts are moved to a separate queue from the queue for successfully completed parts. The Petri net model of the plant is shown in figure 1, and a description of the various places and transitions is given in table 1.

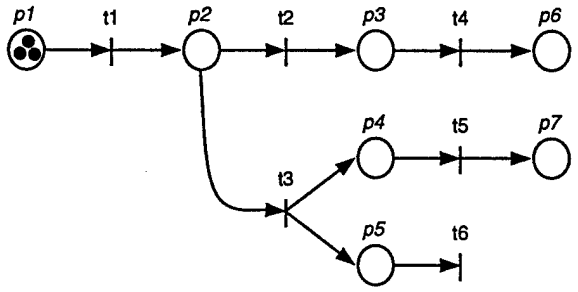


Figure 1: Petri net model of an uncontrolled unreliable machine.

The plant model has two uncontrollable transitions, t_2 and t_3. Transition t_3 represents machine break down and so obviously can not be controlled. Transition t_2 is considered uncontrollable because the controller can not force the machine to instantly finish a part that is not yet completed, nor does it direct the machine to stop working on an unfinished part. The transition is labeled uncontrollable in order to prevent a control design from attempting either of these two actions.

Places	
p_1	Input queue – Number of parts remaining
p_2	Machine is busy, part is being processed
p_3	Waiting for transfer to completed parts queue
p_4	Waiting for transfer to damaged parts queue
p_5	Machine is waiting to be repaired
p_6	Completed parts queue
p_7	Damaged parts queue
Transitions	
t_1	Part moves from input queue to machine
t_2	*Uncontrollable:* Part processing is complete
t_3	*Uncontrollable:* Machine fails, part is damaged
t_4	Part moves to completed parts queue
t_5	Part moves to damaged parts queue
t_6	Machine is repaired

Table 1: Place and transition descriptions for the Petri net of figure 1.

4.1 Controller Synthesis

The Petri net model of the plant has the following incidence matrix and marking vector.

$$D_p = \begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 & -1 & 0 \\ 0 & 0 & 1 & 0 & 0 & -1 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & \underbrace{0 \quad 0}_{D_{uc}} & & 0 & 1 & 0 \end{bmatrix} \quad \mu_p = \begin{bmatrix} \mu_1 \\ \mu_2 \\ \mu_3 \\ \mu_4 \\ \mu_5 \\ \mu_6 \\ \mu_7 \end{bmatrix} \tag{19}$$

The initial conditions are $\mu_{p_0} = \begin{bmatrix} 3 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T$.

If the machine is broken, we do not want to load a new part until repairs have been completed. This means that places p_2 and p_5 should contain at most one token:

$$\mu_2 + \mu_5 \leq 1 \tag{20}$$

Parts waiting to be transferred to a storage queue, whether completed or damaged, wait in the same position on the machine. The Petri net model uses two places, p_3 and p_4, to

represent waiting parts, because there are two different destinations. In order to prevent conflict, the second constraint is

$$\mu_3 + \mu_4 \leq 1 \tag{21}$$

Using the matrix form of constraint (1) we have

$$\underbrace{\begin{bmatrix} 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 \end{bmatrix}}_{L} \mu_p \leq \underbrace{\begin{bmatrix} 1 \\ 1 \end{bmatrix}}_{b} \tag{22}$$

First we must check the uncontrollability condition.

$$LD_{\text{uc}} = \begin{bmatrix} -1 & 0 \\ 1 & 1 \end{bmatrix}$$

We need all of the elements of LD_{uc} to be less than or equal to zero if we are to avoid using uncontrollable transitions. There is no problem with the first row, but a transformation will have to be found to eliminate the 1's in the second row. This can be done by applying row operations from the matrix D_{uc} to eliminate the positive elements in the second row of LD_{uc}.

$$\begin{bmatrix} 0 & 0 \\ -1 & -1 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \\ \hline 1 & 1 \end{bmatrix} \quad \begin{array}{c} \text{Row 8} = \text{Row 8} + \text{Row 2} \\ \Longrightarrow \end{array} \quad \begin{bmatrix} 0 & 0 \\ -1 & -1 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \\ \hline 0 & 0 \end{bmatrix}$$

Because constraint (20) required no transformation, the first row of R_1 will be all zeros. A row operation involving the addition of the second row of the D_{uc} matrix is required to transform constraint (21), thus the second row of R_1 will be all zeros with a one in the second column. It was not necessary to premultiply either constraint, thus R_2 will be an identity matrix.

$$R_1 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \qquad R_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

We now apply equations (14) and (15) to find the transformed constraints represented by L' and b'.

$$\underbrace{\begin{bmatrix} 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 \end{bmatrix}}_{L'} \mu_p \leq \underbrace{\begin{bmatrix} 1 \\ 1 \end{bmatrix}}_{b'}$$

The controller is the calculated using equations (17) and (18).

$$D_c = -L'D_p = \begin{bmatrix} -1 & 1 & 0 & 0 & 0 & 1 \\ -1 & 0 & 0 & 1 & 1 & 0 \end{bmatrix}$$

$$\mu_{c_0} = b' - L'\mu_{p_0} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

The controlled net is shown in figure 2. The constraint logic is enforced and no input arcs are drawn to the uncontrollable transitions.

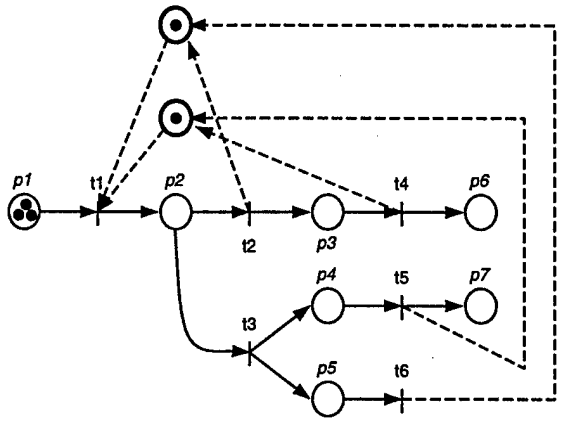


Figure 2: The controlled unreliable machine.

4.2 Discussion

An extensive look at many of the issues central to this research can be found in the work of Li and Wonham [3]. These authors show that optimal, or maximally permissive, control actions which account for uncontrollable transitions can be found by repeated applications of a linear integer programming problem (LIP), assuming that valid control actions actually exist and that the uncontrollable portion of the net contains no loops. They also give sufficient conditions under which the solution to the LIP has a closed form expression. These conditions place a certain tree structure on the uncontrollable portion of the net. When this tree structure is further limited, Li and Wonham are able to prove that the optimal control law which insures $L\mu_p \leq b$ can be written $C\mu_p \leq d$. This is the case where it is possible to represent the action of the optimal control law with ordinary Petri nets. In this situation, it is possible to find R_1 and R_2 by performing row operations on $\begin{bmatrix} D_{uc} \\ LD_{uc} \end{bmatrix}$ which is much more desirable, computationally, than analytically solving an LIP. However the tree structure assumed by Li and Wonham is only sufficient, not necessary, for example, the structure of the uncontrollable part of the plant in section 4 does not conform to Li and Wonham's "type 2 tree structure," however an optimal solution was found and implemented using an

ordinary Petri net controller. There are also cases where, following the procedures presented above, suboptimal Petri net controllers may be derived. These suboptimal controllers may be sufficient for many tasks, depending on the application.

5 Conclusions

This paper has presented computationally efficient methods for constructing feedback controllers for ordinary Petri nets, even in the face of uncontrollable and unobservable plant transitions. The method is based on the idea that specifications representing desired plant behaviors can be enforced by making them invariants of the controlled net, and that simple row operations on a matrix containing the uncontrollable and unobservable columns of the plant incidence matrix can be used to eliminate controller use of illegal transitions.

The significance of this particular approach to Petri net controller design is that the control net can be computed very efficiently, thus the method shows promise for controlling large, complex systems, or for recomputing the control law online due to some plant failure.

There are several areas of ongoing research for this work. Necessary and sufficient conditions for a linear control law to be maximally permissive in the face of uncontrollable and unobservable transitions are not known. Time is becoming an increasingly important factor in the area of DES control. Ordinary Petri nets are sufficient for modeling sequences in time and concurrency, but it may be desirable to extend the method for use with actual timed Petri nets. It may also be possible to extend the applicability of the method by expanding the kinds of constraints that may be enforced. Methods for transforming logical predicates on the plant behavior into linear inequality constraints, and a class of nonlinear constraints, are currently being explored.

References

[1] A. A. Desrochers and R. Y. Al-Jaar, *Applications of Petri Nets in Manufacturing Systems*, IEEE Press, Piscataway, NJ, 1995.

[2] A. Giua, F. DiCesare, and M. Silva, "Generalized mutual exclusion constraints on nets with uncontrollable transitions", In *Proceedings of the 1992 IEEE International Conference on Systems, Man, and Cybernetics*, pp. 974–979, Chicago, IL, October 1992.

[3] Y. Li and W. M. Wonham, "Control of vector discrete event systems II – controller synthesis", *IEEE Transactions on Automatic Control*, vol. 39, no. 3, pp. 512–530, March 1994.

[4] J. O. Moody and P. J. Antsaklis, "Supervisory control of Petri nets with uncontrollable/unobservable transitions", In *Proceedings of the 35th IEEE Conference on Decision and Control*, pp. 4433–4438, Kobe, Japan, December 1996.

[5] J. O. Moody and P. J. Antsaklis, "Characterization of feasible controls for Petri nets with unobservable transitions", In *Proceedings of the 1997 American Control Conference*, volume 4, pp. 2354–2358, Albuquerque, New Mexico, June 1997.

[6] J. O. Moody, P. J. Antsaklis, and M. D. Lemmon, "Automated design of a Petri net feedback controller for a robotic assembly cell", In *Proceedings of 1995 INRIA/IEEE Symposium on Emerging Technologies and Factory Automation*, volume 2, pp. 117–128, Paris, France, October 1995.

[7] J. O. Moody, P. J. Antsaklis, and M. D. Lemmon, "Feedback Petri net control design in the presence of uncontrollable transitions", In *Proceedings of the 34th IEEE Conference on Decision and Control*, volume 1, pp. 905–906, New Orleans, LA, December 1995.

[8] T. Murata, "Petri nets: Properties, analysis, and applications", *Proceedings of the IEEE*, vol. 77, no. 4, pp. 541–580, 1989.

[9] J. L. Peterson, *Petri Net Theory and the Modeling of Systems*, Prentice Hall, Engelwood Cliffs, NJ, 1981.

[10] P. J. G. Ramadge and W. M. Wonham, "The control of discrete event systems", *Proceedings of the IEEE*, vol. 77, no. 1, pp. 81–97, 1989.

[11] W. Reisig, *Petri Nets*, Springer-Verlag, Berlin; New York, 1985.

[12] K. Yamalidou and J. C. Kantor, "Modeling and optimal control of discrete-event chemical processes using Petri nets", *Computers in Chemical Engineering*, vol. 15, no. 7, pp. 503–519, 1991.

[13] K. Yamalidou, J. O. Moody, M. D. Lemmon, and P. J. Antsaklis, "Feedback control of Petri nets based on place invariants", *Automatica*, vol. 32, no. 1, pp. 15–28, January 1996.

APPENDIX I

J.O. Moody and P.J. Antsaklis
"Deadlock Avoidance in Petri Nets with Uncontrollable Transitions"
ISIS Technical Report isis-97-016, October 1997.

Deadlock Avoidance in Petri Nets with Uncontrollable Transitions

Technical Report of the ISIS Group
at the University of Notre Dame
ISIS-97-016
October, 1997

John O. Moody and Panos J. Antsaklis
Department of Electrical Engineering
University of Notre Dame
Notre Dame, IN 46556

Interdisciplinary Studies of Intelligent Systems

Deadlock Avoidance in Petri Nets with Uncontrollable Transitions[1]

Abstract

Recent results in the literature have provided efficient control synthesis techniques for the problem of deadlock avoidance in Petri nets. These results are shown to fit within an established framework for the enforcement of linear constraints on the marking behavior of a net. Framing the problem in this way allows uncontrollable transitions to be included in the plant model when deadlock avoidance is performed. This method for constructing deadlock avoiding supervisors in the face of uncontrollable transitions is described and illustrated with an example.

1 Introduction

Deadlock avoidance is an important and difficult problem in the area of supervisory control. When the discrete event system (DES) is modeled by a Petri net (PN) [15–17], the net is said to be *deadlocked* if no transition in the net is able to fire. A net is called *live* if every transition can be, eventually, fired again and again. A deadlock-free net may not necessarily be completely live.

An efficient supervisory control technique for deadlock avoidance in Petri nets has been proposed in [4]. This method will guarantee deadlock-freedom, and also liveness for a large, useful class of Petri nets. The procedure does not account for possibly uncontrollable transitions within the plant, but uncontrollable events are a standard feature in most supervisory control frameworks.

Techniques for enforcing general linear constraints on Petri nets with uncontrollable transitions do exist [8, 9, 11–13]. These controllers enforce linear inequalities on the reachable markings of the plant while avoiding the inhibition of uncontrollable transitions. Unfortunately these controllers have not, in the past, accounted for the deadlock problem. In fact, the supervisors generated by these techniques may actually be the cause of plant deadlocks!

In this paper, the results for deadlock avoidance are placed within the framework for constraint enforcement in the face of uncontrollable transitions. The combined technique expands the applicability of both control procedures, adding the ability to handle uncontrollable transitions to one, and the avoidance of deadlock to the other.

Section 2 summarizes the Petri net concepts of traps and siphons, which are important to the understanding of the deadlock avoidance procedure discussed in the section 3. Section 4 then shows how this procedure can be placed within an established framework for the

[1]This research was partially funded by the National Science Foundation. Grant ECS95–31485.

supervisory control of Petri nets with uncontrollable transitions. The technique is illustrated in section 5 using an unreliable machine serviced by automated guided vehicles. Concluding remarks appear in section 6.

2 Petri Net Siphons and Traps

Traps and siphons[2] (see [5, 15, 17]) are sets of Petri net places. *Once the set of places in a trap become marked, the trap will always be marked for all future reachable markings. Similarly, once the marking of a siphon becomes empty, the siphon will remain empty.*

Traps and siphons are defined by the nature of the input and output transitions into a given set of places. Let $\bullet p$ refer to the set of input transitions into the place p, and let $p\bullet$ refer to the set of output transitions from the place p. The "bullet" notation can also be used with sets of places. If S is a set of places, then $\bullet S$ and $S\bullet$ refer to the set of input and output arcs into the entire set S.

Definition 1. A set of places S is a **siphon** iff

$$S\bullet \subseteq \bullet S$$

S is a **minimal siphon** iff there does not exist another siphon P such that $P \subset S$.

The definition of a trap is similar.

Definition 2. A set of places S is a **trap** iff

$$\bullet S \subseteq S\bullet$$

S is a **minimal siphon** iff there does not exist another siphon P such that $P \subset S$.

A place invariant vector with nonnegative elements indicates a set of places that is both a trap and siphon. The converse does not hold true. For example, consider a net with no source or sink transitions, i.e., all transitions have both input and output places. By definition, the set of all the net's places is both a trap and a siphon, however such a net may not be covered by a place invariant.

[2]Siphons are sometimes called "deadlocks."

3 Deadlock Avoidance

3.1 Structural Conditions for Liveness

Siphons are of particular interest in the area of deadlock avoidance; once a siphon becomes emptied of tokens, it will forever remain empty and all of the transitions that receive input arcs from these places will be dead. Barkaoui and Abdallah [4] have introduced the notion of a controlled siphon.

Definition 3. For a Petri net with initial marking μ_0, a **controlled siphon** is a siphon that remains marked for all markings reachable from μ_0.

A controlled siphon may be either trap-controlled or invariant-controlled. A *trap-controlled siphon* contains a trap that is initially marked, thus preventing the siphon from ever losing all of its tokens. An *invariant-controlled siphon's* marking is guaranteed by the presence of a place invariant. If the constant weighted sum of markings indicated by a place invariant insures that a siphon will never lose all of its tokens, then that siphon is invariant-controlled. The exact conditions that such a place invariant must meet appear in [4]. An example of an invariant-controlled siphon appears in the example below.

Example. The Petri net of Figure 1 contains two siphons, $S_1 = \{p_2, p_3\}$ and $S_2 = \{p_2, p_3, p_4\}$, where S_1 is the only minimal siphon since $S_1 \subset S_2$. The net contains no traps, so clearly the minimal siphon is not trap-controlled, however an analysis of the net's behavior reveals that this siphon will never be emptied.

The net contains a single place invariant:

$$\mu_2 + \mu_3 - \mu_4 = 1$$

where μ_i is the marking of place p_i. Thus $\mu_2 + \mu_3 \geq 1$ is always true and the siphon is invariant-controlled.

The following propositions relate controlled siphons to deadlock freedom and liveness. The results are well known in the literature, see [5,15,17], however Barkaoui *et al.* [3,4] are responsible for the extension of the results to include the idea of invariant-controlled siphons as well as trap-controlled.

Proposition 4. **Deadlock condition.** A deadlocked Petri net contains at least one empty siphon.

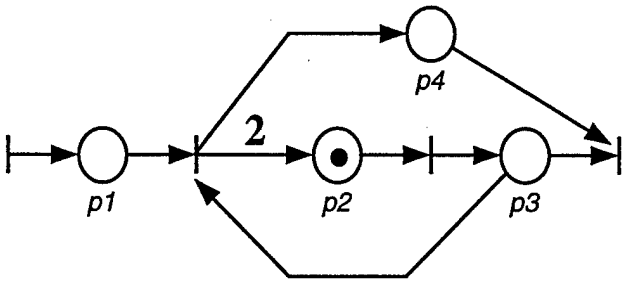


Figure 1: A Petri net with a controlled siphon but no trap.

Proposition 5. **Deadlock-freedom.** A Petri net is *deadlock-free* if every siphon in the net is a controlled siphon.

Proposition 6. **Commoner's Theorem.** An extended free choice (EFC) Petri net is *live* if and only if every siphon in the net is trap-controlled. (Invariant-controlled siphons are not required to insure the liveness of EFC nets.)

Proposition 7. **Liveness for AC nets.** An asymmetric choice (AC) Petri net is live if and only if every siphon in the net is a controlled siphon.

Nonminimal siphons always contain at least one minimal siphon, so it is only necessary to examine a net's minimal siphons when applying the propositions above.

The Petri net of Figure 1 is live. This follows from Proposition 7 since it is an AC net and its single minimal siphon is invariant-controlled.

There are other Petri net classes for which the condition that all siphons are controlled is sufficient for demonstrating that the Petri net is live. These include linear manufacturing lines [10], and production Petri nets [2]. The presence of controlled siphons is then sufficient to insure liveness for a wide variety of Petri nets, and will, at the very least, insure freedom from complete deadlock for Petri nets outside this class.

3.2 Deadlock Avoidance through Supervisory Control

A supervisory control technique is introduced in [4] for handling the problem when not all of the siphons in a given Petri net are controlled. The method involves adding a place for each uncontrolled siphon in the net such that they become controlled. These controller places act to restrict behaviors in the original plant that would lead to deadlock, thus they play the part of a supervisory controller: allowing the plant's state to evolve unrestricted except to prevent transition firings that lead to "forbidden states." An outline of this technique is described below.

Given a conservative, well-marked Petri net with uncontrolled siphons, for each uncontrolled siphon S, create a control place c such that

$$\begin{aligned} c\bullet &= \{t \in S\bullet : |\bullet t \cap S| > |t \bullet \cap S|\} \\ \bullet c &= \{t \in \bullet S : |t \bullet \cap S| > |\bullet t \cap S|\} \end{aligned} \tag{1}$$

where the notation $|x|$ refers to the number of elements in the set x, and the weights of the arc transitions are given by the differences $|\bullet t \cap S| - |t \bullet \cap S|$ and $|t \bullet \cap S| - |\bullet t \cap S|$ for the controller place's output and input arcs respectively. The initial marking of the control place, μ_{c_0}, is given by

$$\mu_{c_0} = \sum_{p_i \in S} \mu_{i_0} - 1 \tag{2}$$

where μ_{i_0} is the initial marking of place p_i in the plant.

Each control place insures that its siphon will never be emptied of all of its tokens. An analysis of the synthesis technique, (1) and (2), shows that this is done by creating place invariants in the controlled Petri net. For each control place c, associated with siphon S, the following place invariant is established in the controlled Petri net.

$$\sum_{p_i \in S} \mu_i - \mu_c = 1 \tag{3}$$

Thus the synthesis technique causes formerly uncontrolled siphons in the plant net to become invariant-controlled siphons in the controlled net.

Example. The free choice Petri net of Figure 2 is conservative: it contains a covering place invariant $x = [1\ 1\ 1]^T$. The three places form a trap, and this trap is minimal. The net contains three siphons:

$$\begin{aligned} S_1 &= \{p_1, p_2\} \\ S_2 &= \{p_1, p_3\} \\ S_3 &= \{p_1, p_2, p_3\} \end{aligned}$$

Place p_1 is marked with two tokens, and it is involved in all three of the net's siphons, thus the net is well-marked.

Siphon S_3 is both trap and invariant-controlled, however it is not minimal. Neither minimal siphon, S_1 and S_2, is controlled. It is easy to see that if the two tokens in p_1 were to both transfer to p_2 or p_3, then deadlock would result.

A deadlock-avoiding controller is constructed according to (1) and (2). Control places c_1 and c_2 are associated with siphons S_1 and S_2 respectively. The controlled system, shown in Figure 3, is live. Note that the resulting net is no longer a free choice net, though it is asymmetric choice.

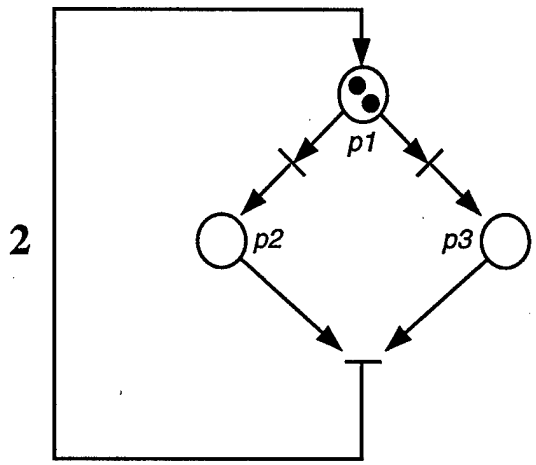


Figure 2: A free choice Petri net with uncontrolled siphons.

Controlling all of the formerly uncontrolled siphons in a net is sufficient for insuring liveness for a wide variety of Petri nets, as noted above. Liveness is not guaranteed for nets outside this class, though they will still be deadlock-free (weakly-live). For these nets, it is proved that if the net is not live (at least one transition is "dead"), then the marking of at least one control place must be zero. Based on this fact, an algorithm is presented in [4] that determines which transitions should fire in order to cause the filling of control places in the fewest number of steps. Because this algorithm actively seeks transitions that should fire, rather than the simple enabling and disabling of transitions of a supervisory controller, it is not discussed here.

4 Handling Uncontrollable Transitions

The deadlock avoidance technique of the previous section is extended here to include nets that contain *uncontrollable transitions*. The firing of an uncontrollable transition is restricted only by the state of the plant, a supervisor can not disable an otherwise enabled uncontrollable transition. Controllers constructed using the technique of the previous section may not be valid if a control place draws an arc to an uncontrollable transition, since this is the mechanism by which the controller inhibits transition firing. This problem is handled by first noting the similarity of the deadlock avoiding controllers of [4] to similar PN supervisory controllers. Methods for handling uncontrollable transitions for these techniques will then be applied to the deadlock avoidance problem.

The controllers in section 3.2 enforce the invariant equation (3). This is equivalent to enforcing the following inequality

$$\sum_{p_i \in S} \mu_i \geq 1 \tag{4}$$

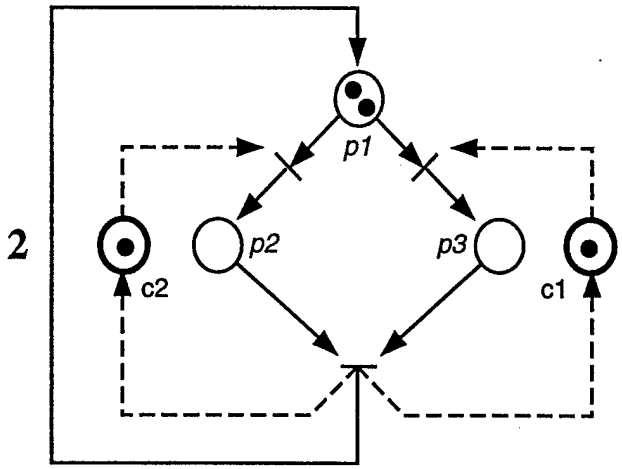


Figure 3: The controller, shown with bold places and dashed arcs, insures the liveness of the FC net of Figure 2.

where μ_c plays the part of a (nonnegative) excess variable. The inequality is intuitively appealing, simply stating that the siphon should never be emptied of tokens. A technique for creating Petri net supervisors for enforcing general linear inequalities on the markings of Petri nets have been in developed in [7, 14, 18]. Furthermore, methods for modifying the inequality such that the resulting controller accounts for uncontrollable transitions have been presented in [8, 9, 11, 13]. The methods of [11, 13, 14, 18] (see also [12], for a summary) are discussed below.

The following constraint is to be imposed on the plant state, $\mu_p \in \mathbb{Z}^n, \mu_p \geq 0$

$$l^T \mu_p \leq b \tag{5}$$

where $l \in \mathbb{Z}^n, b \in \mathbb{Z}$, and $\mathbb{Z}$ is the set of integers. For inequality (4),

$$\begin{aligned} l_i &= \begin{cases} -1 & \text{if } p_i \in S \\ 0 & \text{else} \end{cases} \\ b &= -1 \end{aligned} \tag{6}$$

for $i = 1 \ldots n$, where l_i is the i^{th} element of l. The definition of (6) shows that both sides of inequality (4) were multiplied by -1 to achieve the "less-than-or-equal-to" form of inequality (5).

If all of the transitions within the plant Petri net are controllable and observable, then it has been shown that (5) can be enforced with maximal permissivity by a Petri net controller that produces a place invariant on the closed loop plant-controller system.

The incidence matrix of the closed loop system, D, and its marking, μ, are given by

$$D = \begin{bmatrix} D_p \\ D_c \end{bmatrix} \quad \mu = \begin{bmatrix} \mu_p \\ \mu_c \end{bmatrix} \tag{7}$$

where $D_p \in \mathbb{Z}^{n \times m}$ is the incidence matrix of the plant, $D_c \in \mathbb{Z}^{1 \times m}$ is the incidence matrix of the controller, and $\mu_c \in \mathbb{Z}, \mu_c \geq 0$ is its marking. The controller and its initial marking μ_{c_0} is calculated using

$$D_c = -l^T D_p \tag{8}$$

$$\mu_{c_0} = b - l^T \mu_{p_0} \tag{9}$$

where $\mu_{p_0} \in \mathbb{Z}^n$ is the initial marking of the plant.

Some sets of constraints can not be enforced and thus appropriate controllers do not exist. It is possible to enforce the set of constraints (5) iff

$$b - L\mu_{p_0} \geq 0 \tag{10}$$

Inspection of the constraint definition given by (6) combined with the controller construction of (8) and (9) shows that this controller is identical to the controller constructed using (1) and (2) of section 3.2. Both methods assume that the plant transitions to which the controller arcs are directed are controllable. However, it is possible to transform the original constraint into a new constraint that will result in a controller that does not interfere with uncontrollable transitions while still insuring that the original inequality is maintained.

Negative numbers in the controller incidence matrix D_c correspond to arcs from the control place to the plant transitions. These are the arcs that must be restricted if some transitions are uncontrollable. If the columns of D_c associated with uncontrollable transitions all have nonnegative values, then the controller will meet the requirement imposed by uncontrollability.

Let D_{uc} be an incidence matrix composed of the columns of D_p that correspond to uncontrollable transitions. An examination of equation (8) reveals that the controller will not interfere with the uncontrollable transitions if

$$l^T D_{\text{uc}} \leq 0 \tag{11}$$

where the inequality is read with respect to each element in the vector $l^T D_{\text{uc}}$.

If the inequality is not met, then it is desirable to obtain a new inequality $l'^T \mu_p \leq b'$ such that

1. $l'^T \mu_p \leq b' \rightarrow l^T \mu_p \leq b$
2. $l'^T D_{\text{uc}} \leq 0$

That is, we wish to transform the inequality into a form such that enforcement of the new inequality will also imply the enforcement of the original, while obeying the constraint imposed by the plant's uncontrollable transitions. Analytical and computational techniques for obtaining transformations with these two properties appear in [11–13]. One method of computing the transformation involves performing positive row operations on the matrix $\begin{bmatrix} D_{uc} \\ l^T D_{uc} \end{bmatrix}$ in order to eliminate the positive numbers in the $l^T D_{uc}$ portion of the matrix. This technique is illustrated in the example of the following section.

5 Example – The Unreliable Machine

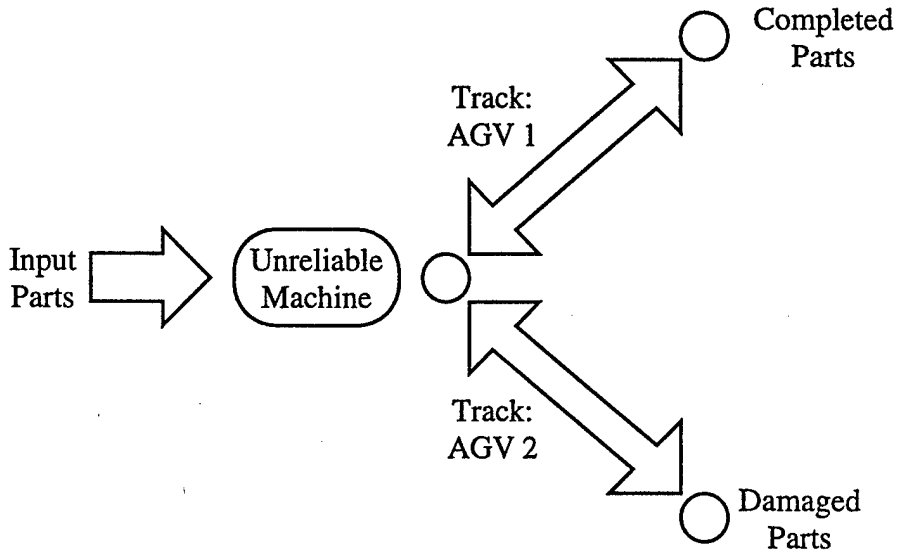


Figure 4: Basic operation of the plant.

The plant of Figure 4 is based around an "unreliable machine" (see [6] and [13]). The machine is used to process parts from an input queue, completed parts are moved to an output queue by an automated guided vehicle (AGV). The machine is considered unreliable because it is possible that it may break down and damage a part during operation. This behavior is captured in the plant model. Damaged parts are moved to a separate queue by a second AGV.

The Petri net model of the plant is shown in Figure 5, and a description of the various places and transitions is given in table 1. Places p_{11}, p_{12}, and p_{13} are used for supervisory control. They insure the mutual exclusion of certain operations within the plant. There is room for only one of the AGV's to pick up a part (either completed or damaged) from the machine, the exclusivity of p_5 and p_{10} is guaranteed by p_{11}. There is also only room for a single part, either damaged or complete, to wait for pickup by an AGV (places p_1, p_2 and p_6), this is guaranteed by p_{12}. Finally, p_{13} insures that if the machine breaks down, a new part

will not be loaded until repairs are complete. It will be shown below that the combined action of these supervisory mechanisms create the potential for the plant to become deadlocked.

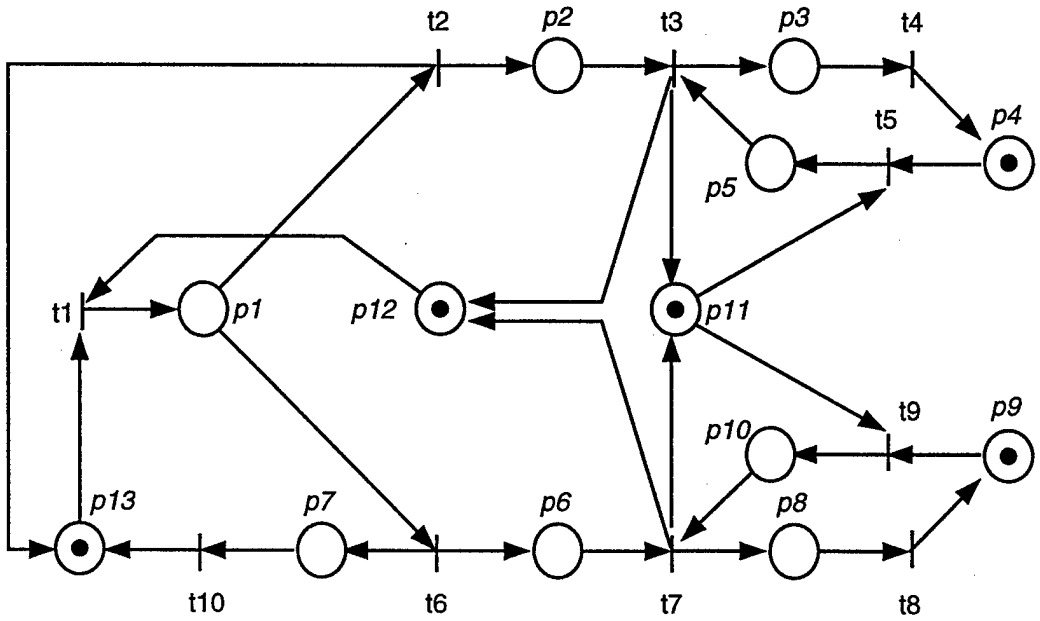


Figure 5: Petri net model of the unreliable machine plant, before accounting for possible deadlock.

The plant model has two uncontrollable transitions, t_2 and t_6. Transition t_6 represents machine break down and so obviously can not be controlled. Transition t_2 is considered uncontrollable because the controller can not force the machine to instantly finish a part that is not yet completed, nor does it direct the machine to stop working on an unfinished part. The transition is labeled uncontrollable to prevent the control design from attempting either of these two actions.

Places	
p_1	Machine is "up and busy," part is being processed.
p_2	Part is waiting for transfer to completed-parts queue.
p_3	Part is being carried to completed-parts queue by AGV 1.
p_4	AGV 1 is free, away from part pick-up position.
p_5	AGV 1 is at pick-up position at machine.
p_6	Part is waiting for transfer to damaged-parts queue.
p_7	Machine is waiting to be repaired.
p_8	Part is being carried to damaged-parts queue by AGV 2.
p_9	AGV 2 is free, away from part pick-up position.
p_{10}	AGV 2 is at pick-up position at machine.
p_{11}	Control: Only one AGV at machine.
p_{12}	Control: Only one completed/damaged part at machine.
p_{13}	Control: Wait for repairs before starting new part.
Transitions	
t_1	Part moves from input queue to machine.
t_2	*Uncontrollable:* Part processing is complete.
t_3	Part is picked up by AGV 1.
t_4	Part is deposited in completed-parts queue by AGV 1.
t_5	AGV 1 moves into pick-up position at machine.
t_6	*Uncontrollable:* Machine fails, part is damaged
t_7	Part is picked up by AGV 2.
t_8	Part is deposited in damaged-parts queue by AGV 2.
t_9	AGV 2 moves into pick-up position at machine.
t_{10}	Machine is repaired.

Table 1: Place and transition descriptions for the Petri net of Figure 5.

The plant has the following incidence matrix.

$$D_p = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & -1 & 0 & 1 & 0 & -1 & 0 \\ -1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (12)$$

There are two uncontrolled siphons in the plant:

$$\begin{aligned} S_1 &= \{p_1, p_2, p_{10}, p_{11}, p_{12}\} \\ S_2 &= \{p_1, p_5, p_6, p_{11}, p_{12}\} \end{aligned}$$

Note that both uncontrolled siphons involve places p_{11} and p_{12}. The interaction of the mutual exclusions being enforced by these two places can result in a deadlock condition. Suppose that AGV 1 moves into position at the machine while it is working on a part, i.e., t_5 fires while $\mu_1 = 1$. Now suppose the machine breaks down (t_6 fires). AGV 1 is stuck waiting for a completed part, and AGV 2 is stuck waiting for AGV 1 to move out of the way: deadlock has occurred. This deadlock condition corresponds to S_1, S_2 corresponds to the analogous situation with the roles of the two machines reversed.

Further supervisors will be added to cause the two siphons to become controlled and thus prevent the possibility of deadlock. For each siphon, a control place will be created that insures that sum of the tokens in the siphon remains greater than or equal to one. Using the notation $l^T \mu_p \leq b$ we have, for S_1,

$$\begin{aligned} l &= \begin{bmatrix} -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 & -1 & 0 \end{bmatrix}^T \\ b &= -1 \end{aligned}$$

Before proceeding to create the control structure using equation (8) and (9), we must check to see if the constraint meets condition (11). The incidence matrix of the uncontrollable portion of the plant, D_{uc}, is given by the second and sixth columns of D_p (since the uncontrollable transitions are t_2 and t_6).

$$l^T D_{\text{uc}} = \begin{bmatrix} 0 & 1 \end{bmatrix}$$

We need all elements of $l^T D_{\text{uc}}$ to be nonpositive to meet inequality (11). If the supervisor were created using the given value of l, then it would attempt to achieve its goal by inhibiting transition t_6, which corresponds to machine break down. Unfortunately the unreliable machine is not impressed by requests from the controller to simply not break, so we will construct a transformed constraint that eliminates the influence of the controller on t_6.

Following the technique of [11, 13], the transformed constraint is constructed by using

positive row operations on $\begin{bmatrix} D_{\text{uc}} \\ l^T D_{\text{uc}} \end{bmatrix}$ to eliminate the positive number in $l^T D_{\text{uc}}$.

$$\begin{bmatrix} -1 & -1 \\ 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ \hline 0 & 1 \end{bmatrix} \quad \Rightarrow \quad \text{Row 14} = \text{Row 1} + \text{Row 14} \quad \Rightarrow \quad \begin{bmatrix} -1 & -1 \\ 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ \hline -1 & 0 \end{bmatrix}$$

Adding row 1 of D_{uc} to eliminate the positive number in $l^T D_{\text{uc}}$ corresponds to adding 1 to the first element of l to construct the new constraint vector l'.

$$\begin{aligned} l' &= \begin{bmatrix} 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 & -1 & 0 \end{bmatrix}^T \\ b' &= -1 \end{aligned}$$

The new constraint, $l'^T \mu_p \leq b'$, represents the following inequality:

$$\mu_2 + \mu_{10} + \mu_{11} + \mu_{12} \geq 1$$

This inequality will insure that the number of tokens in S_1 remains positive and is also admissible with respect to the plant's uncontrollable transitions.

The incidence matrix and initial marking of the control place are now calculated.

$$\begin{aligned} D_c = -l'^T D_p &= \begin{bmatrix} -1 & 1 & 1 & 0 & -1 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \\ \mu_{c1_0} = b' - l'^T \mu_{p_0} &= 1 \end{aligned}$$

The control for siphon S_1 is shown as c_1 in Figure 6. The control, c_2, for siphon S_2 is calculated in a way directly analogous to that of c_1. The supervised plant meets the constraints placed on its behavior and is live.

6 Conclusions

A practical method for deadlock avoidance has been combined with results for enforcing constraints on Petri nets in the presence of uncontrollable transitions. The results expand

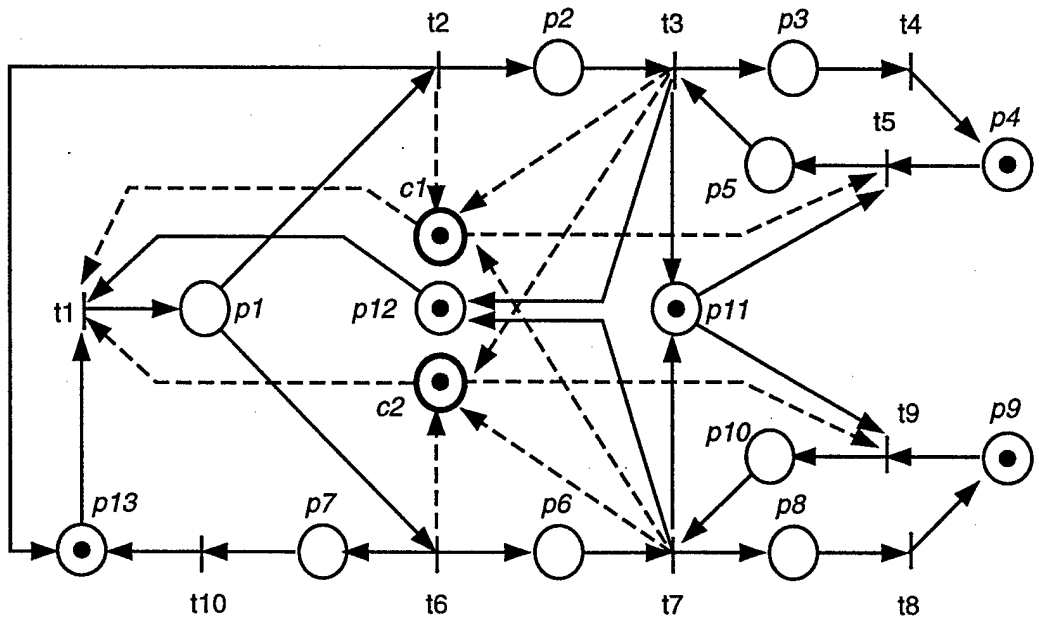


Figure 6: The unreliable machine model is now live.

the applicability and utility of the linear constraint inequality, or "general mutual exclusion constraint", used in [7–9, 11, 18]. Furthermore, they introduce a useful method for dealing with the deadlock problem into the area of PN DES control with its concept of uncontrollable plant transitions.

For some Petri nets, the maintenance of tokens in all of the net's siphons may not be sufficient to guarantee general liveness, though it will at least prevent complete deadlock. In this case, the deadlock avoidance method presented here may form a first step while another control layer actively plans firings to insure that all transitions remain live. This concept fits well with the idea of the "hierarchical intelligent controller" [1]. Here supervision is performed at one layer of the controller, to insure general system and safety constraints, and optimization and planning routines are carried out at another level, working within the boundaries established by the supervisor.

References

[1] P. J. Antsaklis and K. M. Passino, editors, *An Introduction to Intelligent and Autonomous Control*, Kluwer Academic Publishers, 1993.

[2] Z. A. Banaszak and B. H. Krogh, "Deadlock avoidance in flexible manufacturing systems with concurrently competing process flows", *IEEE Transactions on Robotics and Automation*, vol. 6, no. 6, pp. 724–734, June 1990.

[3] K. Barkaoui, "Liveness of Petri nets and its relations with deadlocks, traps, and invariants", Report 92-06, Laboratoire CEDRIC-CNAM, Paris, France, 1995.

[4] K. Barkaoui and I. B. Abdallah, "Deadlock avoidance in FMS based on structural theory of petri nets", In *IEEE Symposium on Emerging Technologies and Factory Automation*, volume 2, pp. 499–510, Piscataway, NJ, 1995. IEEE.

[5] J. Desel and J. Esparza, *Free Choice Petri Nets*, Cambridge University Press, 1995.

[6] A. A. Desrochers and R. Y. Al-Jaar, *Applications of Petri Nets in Manufacturing Systems*, IEEE Press, Piscataway, NJ, 1995.

[7] A. Giua, F. DiCesare, and M. Silva, "Generalized mutual exclusion constraints on nets with uncontrollable transitions", In *Proceedings of the 1992 IEEE International Conference on Systems, Man, and Cybernetics*, pp. 974–979, Chicago, IL, October 1992.

[8] Y. Li and W. M. Wonham, "Control of vector discrete event systems I – the base model", *IEEE Transactions on Automatic Control*, vol. 38, no. 8, pp. 1214–1227, August 1993, Correction in IEEE TAC v. 39 no. 8, pg. 1771, Aug. 1994.

[9] Y. Li and W. M. Wonham, "Control of vector discrete event systems II – controller synthesis", *IEEE Transactions on Automatic Control*, vol. 39, no. 3, pp. 512–530, March 1994.

[10] T. Minoura and C. Ding, "A deadlock prevention method for a sequence controller for manufacturing control", *International Journal of Robotics and Automation*, vol. 6, no. 3, March 1991.

[11] J. O. Moody and P. J. Antsaklis, "Supervisory control of Petri nets with uncontrollable/unobservable transitions", In *Proceedings of the 35th IEEE Conference on Decision and Control*, pp. 4433–4438, Kobe, Japan, December 1996.

[12] J. O. Moody and P. J. Antsaklis, "Supervisory control using computationally efficient linear techniques: A tutorial introduction", In *Proceedings of 5th IEEE Mediterranean Conference on Control and Systems*, volume Session MP1, Paphos, Cyprus, July 1997.

[13] J. O. Moody, P. J. Antsaklis, and M. D. Lemmon, "Feedback Petri net control design in the presence of uncontrollable transitions", In *Proceedings of the 34th IEEE Conference on Decision and Control*, volume 1, pp. 905–906, New Orleans, LA, December 1995.

[14] J. O. Moody, K. Yamalidou, M. D. Lemmon, and P. J. Antsaklis, "Feedback control of Petri nets based on place invariants", In *Proceedings of the 33rd IEEE Conference on Decision and Control*, volume 3, pp. 3104–3109, Lake Buena Vista, FL, December 1994.

[15] T. Murata, "Petri nets: Properties, analysis, and applications", *Proceedings of the IEEE*, vol. 77, no. 4, pp. 541–580, 1989.

[16] J. L. Peterson, *Petri Net Theory and the Modeling of Systems*, Prentice Hall, Engelwood Cliffs, NJ, 1981.

[17] W. Reisig, *Petri Nets*, Springer-Verlag, Berlin; New York, 1985.

[18] K. Yamalidou, J. O. Moody, M. D. Lemmon, and P. J. Antsaklis, "Feedback control of Petri nets based on place invariants", *Automatica*, vol. 32, no. 1, pp. 15–28, January 1996.